POST-QUANTUM GROUP KEY MANAGEMENT FOR INTERNET OF THINGS (IoTs)

By

FOUZIA SAMIULLAH

DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN,

JANUARY 2025

Copyright Statement

DEDICATION

Dedicated to my parents, husband, supervisors, and friends.

ACKNOWLEDGMENTS

I want to begin by expressing my deepest gratitude to Almighty Allah, who has bestowed upon me innumerable blessings, direction, and grace throughout my efforts to earn my Ph.D. It would have been impossible to accomplish anything without His grace and power.

I extend my deepest and most heartfelt gratitude to my research advisor, Dr. Ming-Lee Gan, whose constant support, encouragement, and insightful guidance have been pivotal throughout my Ph.D. journey. Your dedication to mentoring and your ability to inspire excellence have profoundly shaped my academic and professional growth.

I am immensely indebted to my co-supervisor, Prof. Dr. Sedat Akleylek, whose exceptional mentorship and invaluable contributions have been the cornerstone of this research. Without your deep insights, tireless guidance, and unwavering belief in my potential, this work would not have been possible. Your support has been instrumental at every stage, and I will forever cherish the knowledge and skills I have gained under your mentorship.

I also extend my gratitude to my co-supervisor, Dr. Aun Yichiet, for your consistent guidance, valuable feedback, and support, which have played an essential role in shaping this research.

To my family, I owe my deepest thanks. To my husband, Umer Naeem, for his unconditional support and steadfast belief in me, and to my beloved daughters, Eshaal and Mirha, whose love and smiles inspire me to persevere every day. To my mother, Uzma Samiullah, whose comforting words and encouragement have been my emotional anchor, and to my father, Samiullah, whose motivation has been a guiding force.

A special thanks to my siblings—Abdul Rahman, Laiba, and Ateaqullah Rathore—for their love, encouragement, and unwavering belief in me, which have been a constant source of strength.

To my friend Aqsa Iftikhar, for your steadfast emotional support, and to the Pakistani community in Malaysia, for providing a sense of belonging and solidarity that has made this journey lighter and more meaningful.

Lastly, I am profoundly grateful to the Department of Computer and Communication Technology for fostering a supportive and dynamic research environment. It has been a privilege to be part of this institution, which has greatly enriched my academic experience and achievements.

ABSTRACT

POST-QUANTUM GROUP KEY MANAGEMENT FOR INTERNET OF THINGS (IoTs)

Fouzia Samiullah

The emergence of quantum computing presents a significant challenge to conventional cryptographic methods, highlighting the urgent need to create quantum-resistant solutions. In Internet of Things (IoT) networks, ensuring secure group communication facilitates effective and authenticated interactions among numerous devices. Group Key Management (GKM) is a crucial element in this communication framework, ensuring the safe distribution, updating, and revocation of keys across various devices. Nonetheless, current GKM schemes significantly depend on traditional public-key cryptography, exposing them to potential quantum threats. In response to this challenge, Group Authenticated Key Exchange (GAKE) protocols have surfaced as a viable solution, facilitating the secure formation of shared group keys while ensuring mutual authentication among all involved devices. While significant, the design and practical implementation of post-quantum GAKE protocols remains a complex and underexplored study area.

This study focuses on developing a post-quantum secure GAKE protocol specifically designed for IoT environments. To accomplish this, we thoroughly examine current GAKE protocols, pinpointing their shortcomings regarding computational overhead, communication complexity, and vulnerability to quantum attacks. Drawing from these insights, we introduce an innovative Saber-based GAKE protocol that guarantees security within the Quantum Random Oracle Model

(QROM) framework. The protocol's security has been rigorously demonstrated, confirming its resilience in the face of quantum threats.

To assess the protocol's efficiency and scalability, we implemented Saber-GAKE on a local hardware platform (Intel(R) Core(TM) i7-1165G7 with 16 GB RAM, Ubuntu 22.04), and performed comparative benchmarks against the Compiled Kyber GAKE scheme. Additionally, to evaluate its feasibility in constrained IoT environments, performance estimations were derived using existing Saber implementations for ARM Cortex-M4 and M0 processors. Key metrics such as execution time, memory usage, and communication overhead were considered. These results highlight the protocol's superior efficiency and scalability, particularly in large groups of up to 2000 participants, and its compatibility with resource-constrained devices.

This study enhances post-quantum cryptographic protocols for IoT applications by offering a secure, efficient, and scalable GAKE solution. The Saber-GAKE protocol effectively tackles the increasing security demands of IoT networks while providing a robust solution to the potential threats introduced by quantum computing.

Keywords: post-quantum cryptography; group key management; group authenticated key exchange; Internet of Things; Saber

Table of Contents

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

2-AKE	Two-Party Authenticated Key Exchange
AES	Advanced Encryption Standard
AKE	Authenticated key Exchange
DEM	Data Encryption Mechanism
ECC	Elliptic Curve Cryptography
FQ	Future quantum
GKM	Group Key Management
GAKE	Group Authenticated Key Exchange
IoTs	Internet of Things
IND-CPA	Indistinguishability under Chosen-Plaintext Attack
IND-StAA	Indistinguishability against Strong (Adaptive) Active adversaries
IND-AA	Indistinguishability under Adaptive Chosen-Plaintext Attack
IND-CCA	Indistinguishability under Chosen-Ciphertext Attack
KEM	Key Encapsulation Mechanism
KMS	Key management server
LWE	Learning with Error
LWR	Learning with Rounding
MAC	Message Authentication Code
Mod-LWE	Module Learning with Error
Mod-LWR	Module Learning with Rounding
NTT	Number theoretic transform
NIST	National Institute of Standards and Technology

PPT	Probabilistic Polynomial Time
PRF	Pseudorandom function
PKE	Public Key Exchange
PQ	Post-quantum
PQC	Post-Quantum Cryptography
QROM	Quantum Random Oracle Model
QoS	Quality of service
RLWE	Ring Learning with Errors
RFID	Radiofrequency identification
ROM	Random Oracle Model
SHA	Secure Hash Algorithm
SIS	Short integer solution
SGC	Secure Group Communication
TLS	Transport layer security
WSN	Wireless sensor networks

CHAPTER 1

INTRODUCTION

This chapter lays the groundwork for the investigation by examining the security issues in IoT networks in contrast to conventional client-server frameworks. It emphasizes the rising quantum challenge to cryptography and the urgent requirement for post-quantum solutions to guarantee secure group communication. Next, it presents the driving factors, the central issue, and the inquiries guiding the investigation and the intended goals. Ultimately, it outlines the contributions of this research and presents a summary of the overall structure.

1.1 Research Background:

The Internet of Things (IoT) has become increasingly popular among end users due to its widespread presence and diverse applications. IoT encompasses the interconnection of various items or things, interacting through networks using diverse identification and communication technologies. IoT influences our lifestyle, from how we react to how we behave. IoT devices are progressively assuming a significant role in people's daily lives. IoT devices are tangible, network-connected objects with various shapes and features. Devices connected through the internet are rapidly increasing. IoT is a giant network of connected devices. IoT is the interconnection of objects (things) that communicate through networks using various identifying and communication technologies. Furthermore, increasing IoT applications involving group communication influence various important areas of our daily lives. Smart factories, remote healthcare [1], smart homes [2], smart mobility, traffic management, smart grid, transportation systems, logistics [3]and other areas are some examples.

IoT is a sophisticated system that can connect with various technologies, including cloud computing, fog computing, radio frequency identification (RFID), and wireless sensor networks (WSN). Its purpose is to exchange sensory data and enable autonomous control of devices, with or without human involvement [4]. Due to its inherent potential, this technology has experienced rapid growth in various use cases and application domains. In addition to this, new 5G technology significantly speeds up data transfer and enables further scaling of the connectivity process [3]. The deployment of 5G will result in faster broadband speeds and more reliable mobile networks, as well as a faster pace of progress in smart cities, smart vehicles, and smart manufacturing. These advancements open new opportunities for a wide range of applications involving multiple communicating parties. To fully realize the potential of the IoT, experts worldwide emphasize the need for a network architecture that prioritizes security, privacy, and trust [5].

However, the rapid growth of IoT has introduced noteworthy security challenges. The heterogeneous nature of IoTs, with varying capabilities and security features, makes implementing a uniform security policy difficult. As the IoT expands, ensuring security across all devices becomes more and more challenging. Several IoT devices face computational power, memory, and battery life constraints, which can limit the ability to implement strong security measures. In fast growing IoT devices dealing with sensitive data e.g., monitoring patient health condition [4], safe communication is our main concern. As a result, users must maintain control over their data and restrict access to it. Unfortunately, in the past, companies developing IoT devices frequently failed to address this need for security and privacy [5]. IoT devices were commonly deployed without due consideration for security. This resulted in 2016's greatest Distributed Denial of Service (DDoS) attack, which was carried out by thousands of hijacked IoT devices transformed into a botnet to bring down major Internet services such as Netflix and Spotify[6]. In addition, IoT devices frequently deal with sensitive personal information, which gives rise to essential

privacy concerns. Dealing with the challenge of establishing secure communication between devices from various manufacturers and with diverse protocols certainly adds an extra layer of complexity.

The consequences of security breaches in IoT networks can be significant, resulting in the theft of data, manipulation of devices, disruption of networks, financial losses, and damage to reputation. IoT is a heterogeneous interconnection of smart devices across various application domains. The availability of high-speed Internet connectivity alongside complementary advanced technologies such as Big Data [7], Cloud Computing, and easily accessible, inexpensive electronic devices equipped with new wireless communications standards are responsible for the explosive growth of the number of Internet-connected "things." These exponentially increasing numbers of connected smart devices also contribute to the Internet's enormous daily data traffic, data storage capacity, and data availability. Therefore, we, the individuals who incorporate IoT into our residences and businesses, should be more concerned about security. As the attack surface is so vast, it is nearly impossible to provide complete security for IoT infrastructure due to its extensive coverage across numerous application domains and large number of heterogeneous devices. Multiple aspects of IoT security have matured, including privacy, authentication, trust, and communications.

1.1.1 Comparison of IoTs Network with Traditional Client-Server Architectures:

In contrast to conventional client-server architectures that depend on robust servers to manage intensive computational workloads for less capable clients, IoT networks adopt a significantly more decentralized approach, necessitating that numerous devices interact and collaborate with limited central management[8]. The primary differences between IoT and conventional networks are as follows:

a) **Architecture:** In traditional setups, the design typically revolves around a centralized structure, with a server managing most processing, data storage, and security responsibilities. Conversely, IoT networks exhibit greater decentralization, where devices communicate peer-to-peer or connect via local hubs. IoT networks typically take the form of mesh or ad-hoc configurations, necessitating distinct communication and security frameworks compared to conventional client-server architectures.

b) **Asynchronous Data Flow:** In conventional networks, communication typically occurs synchronously, characterized by a direct request-response model (for instance, web servers addressing client requests). In IoT, devices function on asynchronous timelines, sending data at regular intervals or responding to events. The inconsistent data flow presents challenges for managing group keys and synchronizing security updates, given that not all devices might be online or operational simultaneously.

c) **Power and Energy Limitations:** IoT devices generally operate with low power and often require the ability to function for extended periods of battery energy. In contrast, conventional server-based networks depend on robust, high-capacity infrastructure, enabling them to manage more demanding tasks like encryption and data processing without prioritizing energy efficiency.

d) **Memory and Processing Capacity:** Conventional network devices, including PCs and servers, possess significant memory and processing capacity, effortlessly managing intricate cryptographic algorithms and substantial data volumes. IoT devices face challenges due to their limited memory and processing capabilities, which complicate the implementation of conventional security protocols intended for environments with abundant resources. This highlights the necessity for efficient cryptographic solutions in IoT networks that can function effectively within these limitations.

1.1.2 The Quantum Threat to Cryptography and the Need for Post-Quantum Solutions

In the realm of cryptography, the introduction of quantum computing poses a challenge that has the potential to impact the field completely. Quantum computers, in contrast to classical computers, which process data in binary, which can only be either 0s or 1s, make use of quantum bits, also known as qubits. These qubits, due to the phenomenon of superposition, are capable of simultaneously representing both 0s and 1s. Quantum systems can process numerous possibilities concurrently thanks to the superposition characteristic, which results in an exponential increase in the amount of computational power available for solving particular problems. Furthermore, quantum entanglement correlates with the states of qubits, which enables instantaneous coordination and further enhances the efficiency of computer processes.

The computational difficulty of issues such as integer factorization and discrete logarithms is essential to the operation of public-key cryptosystems like RSA and Elliptic Curve Cryptography (ECC). The fact that quantum algorithms, and in particular Shor's algorithm[9], can solve these issues in polynomial time as efficiently as possible, making these techniques susceptible to attack. For instance, Shor's algorithm factors a composite number, which typically takes classical computers exponentially more time to solve than it does today. Similarly, symmetric encryption systems such as AES, despite being more resistant to attacks, are degraded by Grover's algorithm, which in turn diminishes the effectiveness of their security by fifty percent. Since this is the case, a symmetric key of 128 bits in length provides only 64 bits of protection against a quantum attacker, which is why longer key sizes are required[10].

This hazard is especially concerning when considered in the context of Internet of Things networks. Because Internet of Things devices frequently have limited computational power and memory, they rely on lightweight cryptography solutions. To undermine encrypted communications, intercept sensitive data, and forge digital signatures, attackers that are capable

of quantum computing could pose a danger to the security of Internet of Things (IoT) devices. This weakness extends to secure group communication protocols, which could be disrupted by quantum attacks. Examples of such applications include healthcare, smart grids, and intelligent transportation.

Researchers are currently working to develop post-quantum cryptography (PQC), which is a category of cryptographic algorithms meant to withstand both conventional and quantum attacks. This is being done to solve the issues that have been presented. However, in contrast to quantum key distribution (QKD), which necessitates the utilization of specialized quantum hardware, PQC can function on classical systems, which makes it more feasible for wider implementation. Since they are both efficient and resistant to quantum attacks, lattice-based systems have emerged as leading possibilities. The CRYSTALS-Kyber algorithm, which is based on the Module Learning with Errors (MLWE) issue, and the Saber algorithm, which is based on the Module Learning with Rounding (MLWR) problem, are, respectively, examples of these. As part of the National Institute of Standards and Technology's (NIST) attempt to standardize post-quantum cryptography, both systems were highlighted, with Kyber being chosen as the first PQC standard for public-key encryption. Since it is easy to use and suitable for Internet of Things applications, Saber continues to be a powerful solution.

1.1.3 Secure Group Communication:

In the realm of IoTs, the importance of secure group communication cannot be overstated. Secure Group Communication (SGC) is crucial in upholding various security attributes necessary for secure data transmission between IoT devices[11]. In many scenarios, IoT applications necessitate the ability of one device to securely transmit information to several recipients, such as distributing sensor data throughout a smart city network. The requirements of secure group communication

can be divided into two categories: security requirements and efficiency requirements (as shown in Figure 1.1)

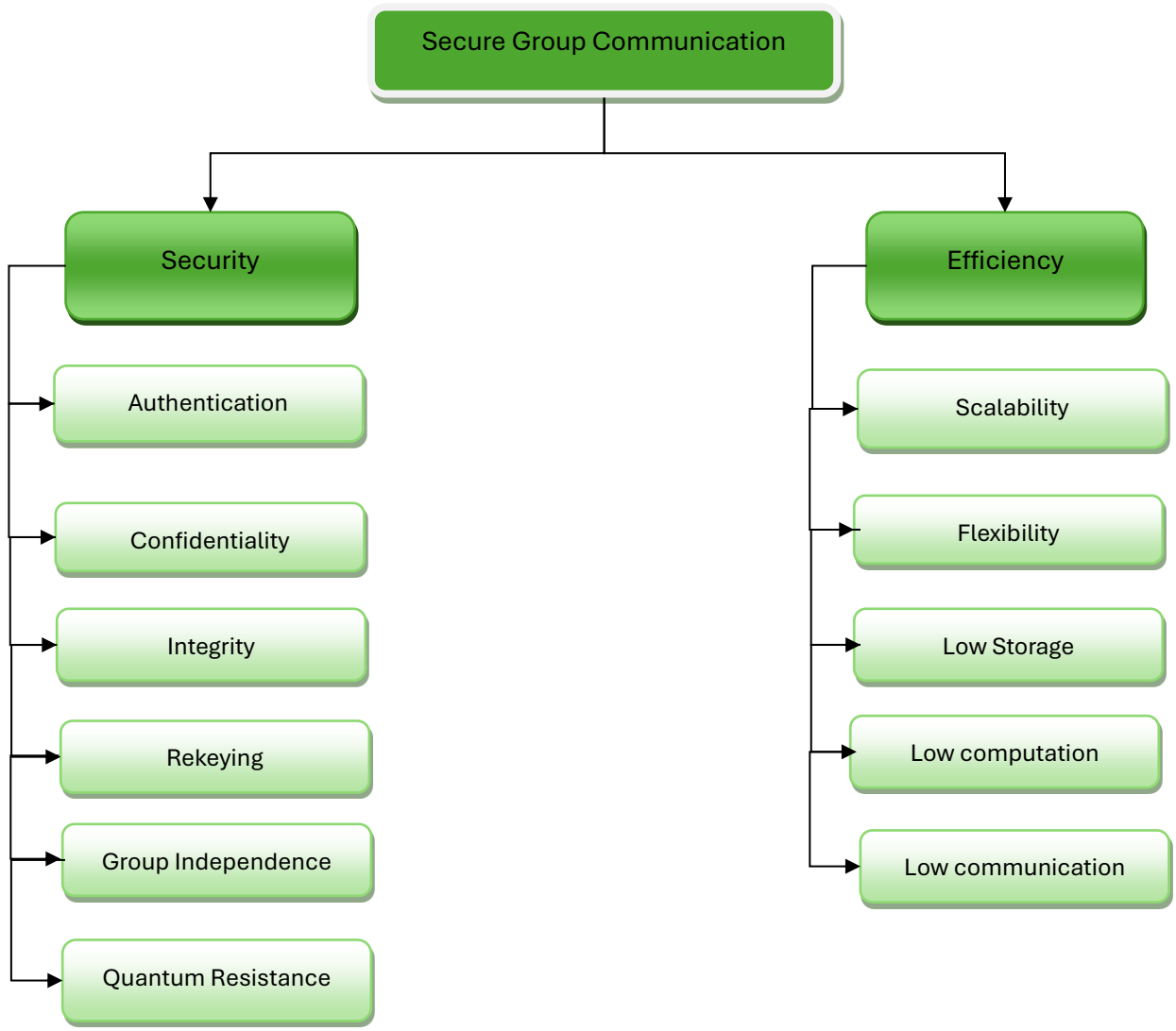

Figure 1. 1 Secure group communication requirements.

1.1.3.1 Security Requirements:

a) **Authentication:** Authentication guarantees that only authorized nodes can access group communication. Nodes are required to authenticate their identities before joining a group by employing methods such as group keys, pairwise keys, or certificates. To prevent identity-

related assaults, participants must be authenticated continuously, which extends beyond entry[12]. There are numerous methods of authentication:

Group Key Authentication: All members utilize a shared secret key. The possession of this key validates membership. Pairwise Key Authentication: Communication is guaranteed to be secure, even in the presence of other members, as unique keys are established between each pair of nodes and Certificates: Digital certificates issued by a reputable Certificate Authority (CA) attach a public key to a node's identity, enabling tamper-resistant, robust authentication. Authentication must be implemented throughout group communications to safeguard against identity spoofing and validate the originator's identity.

b) **Integrity** refers to the correctness and consistency of group messages. Integrity guarantees that group messages are transmitted as intended, free from unauthorized alterations. When a message is conveyed, it must remain unchanged to ensure accuracy and dependability [13].

Hash Functions: Cryptographic hash algorithms, such as SHA-256, provide a unique fixed-length digest of a message to attain integrity. Any alteration in the message results in a distinct hash value, facilitating the verification of message integrity. Digital Signatures By integrating hashing with a sender's private key, digital signatures guarantee integrity and authenticate the sender. Robust Encryption Keys Encryption, like AES, provides a protective barrier by thwarting manipulation during message transmission. These strategies guarantee the integrity and reliability of communication, thwarting threats such as message injection or tampering.

c) **Confidentiality:** Restricting access to group messages and guaranteeing they are solely available to authorized users is crucial. Unsecured communications may result in data breaches or unauthorized access, threatening the group's operations

Confidentiality is attained through Symmetric Encryption: Shared secret keys (e.g., AES) facilitate the encryption and decryption of messages, ensuring that only individuals possessing the key can access the content. Asymmetric Encryption: In situations necessitating safe key exchange or individual-specific encryption, public-key cryptography (e.g., RSA, ECC) is employed. Access Control Policies: Establish precise controls to designate which members are authorized to access specific resources, and Robust key distribution and management systems, such as the Diffie-Hellman exchange or Certificate-Based Key Exchange, are essential for preserving confidentiality in dynamic group memberships [14].

d) **Rekeying:** It refers to updating the session key. Long-term key had more chance to compromise frequently. Every change in membership necessitates the rekeying of associated keys. The group key should be revoked immediately if a member's membership changes. Otherwise, until the group key is updated, the revoked nodes can continue to use the group communication. We modify the encryption key to reduce the amount of data encrypted with the same keys. The techniques for key updates provide options for managing the lifecycle of encryption keys in group communication scenarios. (As shown in Table 1.1)

e) **Group Independence:** In many cases, nodes concurrently engage in different groups. To guarantee secure and efficient functioning, security parameters for each group must stay independent. A breach in one group does not compromise the security of other groups. Each node is required to uphold a profile for each group, which encompasses: A unique group key or collection of keys for encryption and authentication, defined security protocols and parameters for each group, including rekeying intervals and access controls, and details regarding group controllers encompass the addresses or names of the trustworthy people overseeing the group.

The independence of groups guarantees that a compromise in one group does not enable an attacker to exploit other groups.

f) **Quantum Resistant:** The advent of quantum computing presents significant challenges with classical encryption techniques. Algorithms like RSA and ECC, which depend on the complexity of integer factorization and discrete logarithms, are susceptible to quantum assaults utilizing Shor's technique.

To ensure the security of group communication against future threats, implementing post-quantum cryptography algorithms is essential.

Lattice-based cryptography depends on the difficulty of lattice issues, which are resilient to quantum assaults; hash-based cryptography employs hash functions to generate safe digital signatures independent of number-theoretic assumptions. Code-based cryptography is predicated on the challenge of deciphering error-correcting codes, and Hybrid Cryptographic Systems integrate classical and quantum-resistant algorithms to facilitate seamless transitions as quantum technologies advance.

By implementing these measures, SGC systems can guarantee enduring security even in the quantum era.

Table 1. 1 Definitions of different key update strategies

Key update types	Definitions
Periodic	It is a key update strategy in which encryption keys are updated regularly. Significant changes occur on a predetermined schedule, such as hourly, daily, weekly, or any other regular interval. This method ensures that keys are routinely updated, minimizing the risk of long-term compromises and preserving group communication security.
Probabilistic	It is a strategy in which key updates occur randomly according to a probability distribution. Keys are updated at random intervals, adding an aspect of unpredictability rather than according to a

	predetermined schedule or trigger. This approach adds layer of security, making it difficult for potential attackers to predict when critical updates will be implemented.
On-demand	It is a strategy in which significant changes occur in response to specific requests or triggers from group members or the system. Key updates may be initiated when a member requests a new key, identifies a security issue, or meets certain predefined conditions. This method allows for greater flexibility in key management, as updates are implemented as needed rather than according to a predetermined schedule.
Session wise	It is a strategy in which encryption keys are updated at the start of each group session or communication session, a distinct period or phase of group communication. This method guarantees that each session begins with a new set of keys, reducing the potential impact of key compromises or assaults on subsequent sessions.
At-membership change	A key update strategy in GKM requires updating encryption keys when group membership changes. This strategy triggers important updates when members join or leave a group or when the number of members changes.

1.1.3.2 Efficiency Requirements

a) **Scalability:** Secure group communication schemes that provide efficiency and security for small groups should be maintained if the group size becomes larger. Most importantly, membership management algorithms must be efficient so that the group controllers can manage multiple requests simultaneously, e.g., when the user joins or leaves an activity. Delivery of the group key to large groups must be in a reasonable amount of time with a reasonable amount of delay and low computational and communication costs [15].

b) **Flexibility:** SGC schemes should work well in different environments. They should support dynamic behaviour and allow users to be added and removed anytime.

c) **Low Storage, Communication, and computation cost:** Secure group communication schemes should be efficient in storage, communication, and computational cost. IoT devices

are resource constraints that make us focus on these specific limitations. Memory to store keys is limited, so the number of keys used to protect group communication must be low. The computational cost must not be heavy as sensors inherently have low-power CPUs. The component's message exchange rate must be low. To avoid sensor node energy, drain, and thus failure, the SGC scheme must not impose a high communication cost.

1.1.3.3 Group Key Management Primitives:

GKM primitive focuses on Primitive requirements and procedures (as shown in Figure 1.2).

The most significant group key management scheme that is considered compatible should have the following primitive requirements: They are classified into five types: Performance, Security, QoS (Quality of service), Key management server, and Group Members[11].

a) Performance Requirement:

Robustness: GKM protocols should be able to handle dynamic group sizes.

1-Affects-N phenomenon: Multiple group members are affected when a single membership status changes throughout the join/leave procedure, decreasing network communication iterations.

Availability of services: The operation of key management structures throughout the entire multicast session is unaffected by the failure of a single node.

b) Security Requirements:

Forward secrecy guarantees that when a member departs from the group, they are entirely barred from viewing any subsequent group keys or messages. This mitigates the risk of any prospective exploitation of previously acquired credentials. Likewise, backward secrecy prevents new members from decrypting communications transmitted before their membership in the group. These procedures guarantee that access to group communication is entirely restricted to active

members, hence preserving confidentiality and integrity. Robust cryptographic techniques, like periodic rekeying and event-driven key updates, are generally utilised to maintain these security principles.

c) QoS (Quality of service):

In multicast communication, QoS is essential for guaranteeing reliable and efficient data transmission. Minimal packet latency and elevated packet delivery ratios are essential for preserving communication quality. These parameters are especially susceptible to alterations in key management, as frequent key upgrades may result in packet delivery delays. By minimizing critical alterations, GKM methods eliminate jitter and enhance the packet delivery mechanism, ensuring that group communication is uninterrupted and high-quality.

d) Key Management Server:

The key management server (KMS) is essential for efficiently administrating group keys, especially in dynamic group environments. The server must manage a substantial influx of messages and adjust to regular membership fluctuations without sacrificing speed. Considering the time necessary for key encryption and decryption is crucial, as these processes directly influence the overall efficiency of the GKM protocol. Scalable and resilient KMS architecture is essential for accommodating extensive groups and ensuring maximum efficiency during rekeying processes.

e) Group Members:

Group members must be adept at efficiently managing cryptographic keys. This necessitates reducing the number of keys each member must retain and utilize for communication. Reduced key storage enhances memory efficiency and facilitates prompt access to essential keys for secure communication among members. For the key server, sustaining lightweight key

structures improves its ability to facilitate frequent and efficient key changes, particularly in dynamic and large-scale group contexts.

1.1.3.4 Group Key Management Procedures:

The GKM protocol specifies how the group key is generated, distributed, and updated. The most important part of group key management is ensuring the secure and reliable delivery of keying materials to all legitimate members [15]. To do this, efficient key distribution, generation, and updating processes must be implemented. Each of these processes must be considered when designing a key management algorithm in resource-constrained network.

a) Key Generation:

The key generation phase involves creating all the other keys and the group key. This assists the key allocation controller distribute the group key to all genuine receivers.

b) Key Distribution:

Key allocation refers to the reliable, efficient, and secure distribution of keying materials to group members. Because group members in wireless networks may be geographically dispersed or move from one location to another, the most important task in group key management is ensuring that the “group key” is delivered to all legitimate members.

c) Re-keying:

The rekeying process guarantees forward and backward secrecy. The group key and other keys are updated, and updated keys are sent to the group members. Reducing rekeying costs is more critical. Key rekeying is the costliest process because it requires the most computation and communication overhead, requiring more time, energy, and resources to generate and distribute the new key.

Since an IoT network can connect many devices with varying functions, each device may communicate with an undetermined number of other devices. Some messages should be sent to multiple devices simultaneously. When sending messages to multiple recipients, group communication can be used in the network to improve efficiency and communication performance. A group key is distributed among group members to ensure secure group communication [16]. Group key refers to the shared encryption key. It is the key upon which the security of group communication relies entirely. Symmetric encryption algorithms encrypt messages within the multicast group member nodes, but the keys used for these encryption processes play a vital role in group key exchange processes. The group key management mechanism has been employed in several works (architecture of multicast centralized).

Dealing with these challenges necessitates the development of novel approaches that offer strong security while maintaining the effectiveness and scalability of IoT networks. Combining PQC with group authenticated key exchange (GAKE) protocols presents a potential solution for strengthening the security of IoT systems against quantum threats, all while maintaining efficient and secure group communication.

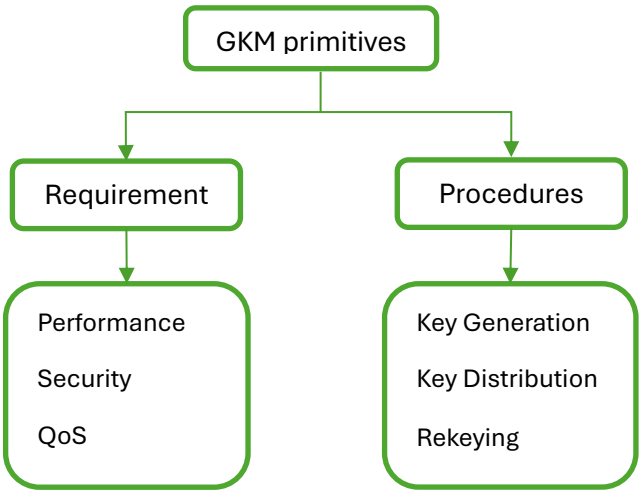

Figure 1. 2 GKM Primitives

Table 1. 2 An Overview of the Selected Definitions of Group Key Management

Definitions of GKM	Source
"The management of secret keys on distributed entities is called GKM"	[17]
"The GKM is the core of secure communication. Its main role is to establish secure links between the group members."	[18]
"GKM enables secure access to relevant information, such as group keys in order to grant confidentiality, integrity as well as sender- and group-authentication."	[19]
"GKM involves the handling, revocation, updating and distribution of cryptographic keys to members of various groups in a communication network"	[20]
"GKM represents the fundamental mechanism for managing the dissemination of keys for access control and secure data distribution."	[1]
"GKM is one of the fundamentals in securing group communications. A group key essentially is a secret key shared by all members of a group so that all group communication packages are encrypted before they are being transmitted using this group key."	[21]

1.2 Motivation and Problem Statement

The rapid development of quantum computing represents a profound shift in the cybersecurity landscape[22]. Quantum computers have the potential to break conventional cryptographic systems that are foundational to current secure communications. Algorithms such as RSA, ECC, and other public-key cryptographic mechanisms could become obsolete due to quantum-enabled computational power. This looming threat underscores the urgent need for cryptographic solutions that can withstand quantum attacks, commonly called post-quantum cryptography (PQC)[23].

As IoT technology expands, its applications become increasingly integral to modern life. IoT devices support critical sectors such as healthcare, smart cities, industrial automation, and logistics. However, these devices often operate with limited computational resources, memory, and energy, posing a unique challenge to implementing complex cryptographic protocols. The constrained nature of IoT environments makes them particularly vulnerable to security breaches if adequate cryptographic defenses are not employed. Therefore, developing cryptographic protocols that are secure against quantum adversaries and optimized for resource-limited environments is essential for IoT systems' long-term viability and security. The primary motivations driving this research are outlined below:

1. **The Impact of Quantum Computing on Cryptography**: The advent of quantum computing is expected to disrupt conventional cryptographic systems. Algorithms like Shor's algorithm render public-key cryptosystems such as RSA and ECC vulnerable[9], necessitating the development of PQC protocols to safeguard data integrity in the IoT landscape.
2. **Limitations of Current PQC Algorithms for IoT Devices**: While PQC algorithms are essential for countering quantum threats, many of these solutions are resource-intensive, requiring significant computational power, memory, and energy. IoT devices, characterized

by their constrained environments, cannot accommodate these demands without substantial performance compromises. Thus, lightweight PQC algorithms must be innovated to ensure security without overwhelming device capabilities[24].

3. **Challenges in Group Key Management (GKM)**: Managing cryptographic keys for secure group communication in dynamic IoT networks is complex. The inherent nature of IoT, where devices frequently join and leave the network, demands scalable and efficient GKM protocols. Existing schemes often fail to balance security and resource efficiency, making them less suitable for large-scale and resource-constrained IoT applications[25].
4. **Integration with Existing IoT Systems**: Introducing new cryptographic measures into established IoT infrastructures without disruption is challenging. Protocols must be secure and seamlessly integrated into current systems to facilitate adoption. Ensuring that quantum-resistant solutions are compatible with existing IoT frameworks is essential for practical deployment[26].

1.2.1 Problem Statement

Despite significant progress in post-quantum cryptography, there is a noticeable gap in the development of Group Authenticated Key Exchange (GAKE) protocols tailored to the needs of resource-constrained IoT networks. While many existing GAKE protocols leverage classical cryptographic techniques, they fall short of offering sufficient security when faced with the potential power of quantum attacks. Moreover, the protocols designed with quantum resistance in mind often do not consider the specific resource limitations characteristic of IoT devices, resulting in impractical implementations that hinder performance and scalability.

The existing body of research highlights several critical shortcomings:

1. **Insufficient Optimization for IoT Constraints:** Most GAKE protocols fail to adequately address IoT devices' low computational power and limited memory. This inadequacy leads to performance bottlenecks and inefficient resource use.
2. **Lack of Quantum Resistance:** Current GAKE protocols do not incorporate post-quantum cryptographic algorithms that ensure security against future quantum threats. Without these safeguards, IoT networks remain susceptible to quantum-enabled breaches.
3. **Absence of Real-World Implementation:** Many proposed quantum-resistant GAKE protocols remain theoretical and have not been rigorously evaluated in practical, real-world scenarios involving IoT devices. This gap leaves uncertainties in their performance and applicability.

The existing GKM protocols used for SGC are focused on either the medium, which is based on pre-quantum cryptography, or a few proposed GKM protocol quantum resistance solutions, none adequately addressed the specific challenges of IoT environments. We aim to bridge these gaps by introducing a novel GAKE protocol utilizing the SABER family of post-quantum cryptographic tools. SABER, known for its balanced trade-offs between security and computational efficiency, provides a promising framework for addressing the dual requirements of quantum resistance and low resource consumption. The primary objective is to design and evaluate a GAKE protocol that ensures robust security against quantum adversaries while maintaining the performance metrics necessary for IoT environments, such as reduced memory usage and faster computation times.

1.3 Research Content

The development of new cryptographic protocols is required to guarantee the security of IoT systems due to the accelerated advancements in quantum computing. In IoT environments, GKM

is essential for facilitating secure group communication, as devices must share data collaboratively while protecting against potential attacks. This study addresses two primary research questions (RQs) by concentrating on creating a post-quantum secure GKM protocol specifically designed to meet the needs of the IoT.

1.3.1 Research Questions

RQ1: What are the existing group key management protocols based on the hardness assumption of post-quantum secure cryptography?

The present research explores a variety of GKM protocols, which encompasses both conventional pre-quantum schemes and those that are founded on post-quantum cryptography. Its objective is to determine their security features, design frameworks, and scenario-specific applications, including intelligent transportation, healthcare systems, and smart infrastructures. The challenges of employing post-quantum solutions in resource-constrained IoT environments and the limitations of pre-quantum schemes in addressing the quantum threat will be underscored by this comprehensive analysis.

RQ2: What are the primitives for designing a novel group key management protocol for a random group of IoT users utilizing post-quantum cryptography, and how secure is it?

Secondly, it aims to create a unique group key management protocol designed explicitly for random groups of IoT users, utilizing post-quantum cryptographic techniques. Lastly, it seeks to thoroughly demonstrate the proposed protocol's post-quantum security, which will ensure its efficiency and durability in protecting group communications in IoT settings.

1.3.2 Objectives of the study

RQ1: The first question being investigated aims to evaluate the present state of GKM protocols that employ the mathematical foundations of post-quantum secure cryptography. This includes an examination of scenario-based GKM schemes, post-quantum advancements, and pre-quantum protocols. The study provides a fundamental comprehension of GKM techniques by examining pre-quantum approaches, including Diffie-Hellman-based GKM and symmetric key distribution methods. Although these methods are effective in classical environments, they are significantly constrained when faced with quantum adversaries.

The emphasis then transitions to post-quantum GKM protocols, which capitalize on cryptographic primitives that are impervious to quantum computing attacks. These protocols are frequently predicated on hardness assumptions, including code-based cryptography, multivariate polynomial problems, and lattice-based problems (e.g., Learning with Errors (LWE) and Learning with Rounding (LWR). Lattice-based cryptography has emerged as a prominent candidate due to its scalability and efficiency in resource-constrained environments.

Furthermore, this investigation assesses the feasibility of these protocols in real-world IoT applications, including smart cities, healthcare systems, and industrial IoT. It identifies obstacles such as computational complexity, scalability, and integration with existing infrastructure. This analysis identifies the deficiencies in current research, thereby facilitating the development of innovative solutions.

RQ2: The second objective of this research is to develop a novel group key management protocol that addresses the distinctive challenges of IoT environments, such as resource constraints and quantum security, by utilizing post-quantum cryptographic primitives, specifically Sabre. The protocol will prioritize the utilization of lattice-based cryptography, specifically the Module Learning with Rounding (MLWR) problem, to optimize security and efficiency while reducing

computational and memory overheads. The protocol's resilience to classical and quantum attacks will be assessed by thoroughly examining its security using formal methods, including the Quantum Random Oracle Model (QROM) and Random Oracle Model (ROM). The performance evaluation will measure execution time, memory utilization, and communication overhead on resource-constrained IoT platforms.

1.4 Novelty and Contribution of the Study

This study makes several impactful contributions to PQC and SGC within IoT environments. The detailed contributions are as follows:

1. **Development of the Post-Quantum GAKE Protocol:** The principal contribution of this research is the design and introduction of Saber-GAKE, a novel group authenticated key exchange protocol grounded in the Module-LWR assumption via the Saber KEM. Unlike existing GAKE protocols, such as Compiled Kyber, which rely on the Module-LWE assumption and are designed for general-purpose applications, Saber-GAKE is specifically evaluated for resource-constrained IoT devices. This work represents the first GAKE construction based on Saber, combining quantum resistance with practical efficiency, and filling a critical gap in the current literature.
2. **Performance Evaluation and Scalability:** The research extensively evaluates the Saber-GAKE protocol, showcasing its performance under various group sizes, including large-scale scenarios with up to 2,000 participants. This performance analysis highlights the protocol's scalability and proves its viability for real-world IoT deployments, where managing secure communications across many connected devices is crucial.
3. **Provision of Security Proof:** The study contributes a formal security proof of the Saber-GAKE protocol based on the Module Learning with Rounding (Module-LWR) assumption within the QROM. This theoretical validation fortifies the credibility of the protocol,

positioning it as a robust solution within post-quantum cryptographic standards and providing a strong foundation for future development.

4. **Mathematical Analysis for IoT Suitability:** The study includes an in-depth mathematical analysis focusing on the Saber-GAKE protocol's resource efficiency, considering computational complexity, memory usage, and cryptographic operations. This analysis confirms that the protocol is suitable for constrained environments like IoT, ensuring minimal resource consumption without compromising security. Such evaluations are critical to prove that the protocol can be effectively deployed in IoT networks where devices operate under stringent resource limitations.

To highlight the novelty of this study within the current research landscape, Table 1.3 presents a comparative summary between the proposed Saber-GAKE and the most closely related work, Compiled Kyber. A more detailed comparison is provided in Chapter 4 and 5.

Table 1. 3 Summary of Novelty – Saber-GAKE vs. Compiled Kyber

	Saber-GAKE	**Compiled Kyber**
KEM	Saber	Kyber
Mathematical Assumption	MLWR	MLWE
Optimization Targets	Evaluated for resource-constrained IoT devices	General-purpose post-quantum GAKE
Security Model	ROM, QROM	ROM, QROM
Correctness and Security Proof	Using Reconciliation conditions and QROM-based	QROM-based

Scalability	Validated up to 2000 participants	Also evaluated up to 2000 participants
Computational Efficiency on ARM (Estimated)	≈ 0.057 ms (M4) ≈ 1.41 ms (M0)	N/A
Memory Usage (Estimated)	≈ 44.1 KB (M4) $\approx 30.95\ KB$ (M0)	N/A
Novel Contribution	First GAKE using Saber with formal proof, tailored for IoT	First implementation using Kyber, not IoT-optimized

1.5 Publications Arising from This Work

The following peer-reviewed publications have arisen from the research presented in this thesis. They reflect the development, analysis, and application of post-quantum group key exchange protocols and secure key management in IoT environments:

- F. Samiullah, M.-L. Gan, S. Akleylek, and Y. Aun, "Quantum Resistance Saber-Based Group Key Exchange Protocol for IoTs," IEEE Open Journal of the Communications Society, vol. 6, pp. 378–398, 2025. doi: 10.1109/OJCOMS.2024.3516005
- F. Samiullah, M.-L. Gan, S. Akleylek, and Y. Aun, "Group Key Management in Internet of Things: A Systematic Literature Review," IEEE Access, vol. 11, pp. 77464–77491, 2023. doi: 10.1109/ACCESS.2023.3298024
- F. Samiullah, M.-L. Gan, S. Akleylek, and Y. Aun, "Quantum Resistance Group Key Management for IoTs," in Proc. 2nd Int. Conf. on Emerging Trends in Electrical, Control,

and Telecommunication Engineering (ETECTE), Lahore, Pakistan, 2023, pp. 1–6. doi: 10.1109/ETECTE59617.2023.10396800

- F. Samiullah, M.-L. Gan, S. Akleylek, and Y. Aun, “Post-Quantum Group Key Management in IoTs,” in Proc. 25th Int. Multitopic Conf. (INMIC), Lahore, Pakistan, 2023, pp. 1–6. doi: 10.1109/INMIC60434.2023.10466001
- F. Samiullah, S. Akleylek, M.-L. Gan, and Y. Aun, “Group Key Management in Resource Constraint Environment: Applications and Use Cases,” International Journal of Advanced Natural Sciences and Engineering Researches, vol. 7, no. 3, pp. 269–278, 2023.

1.6 Structure of the Thesis

The next sections of the thesis are organized as follows: Chapter 2 reviews cryptographic foundations, encompassing symmetric and asymmetric cryptography, IoT security in Quantum World, and an overview of post-quantum cryptographic solutions. It looks at current group key management protocols for IoT networks and assesses their quantum resistance. Chapter 3 provides the mathematical foundation, covering lattice-based problems, the Quantum Random Oracle Model (QROM), transformations among encryption schemes, and Abdalla's Compiler for secure group communication. Chapter 4 comprehensively describes the Saber-GAKE protocol, including its design, mathematical analysis, and security evaluation. Chapter 5 outlines the experimental setup, implementation, and results, demonstrating the protocol's efficiency and appropriateness for IoT environments. Chapter 6 concludes the thesis by summarizing the findings and outlining potential future research directions.

CHAPTER 2

LITERATURE REVIEW

This chapter comprehensively reviews cryptographic fundamentals and their applications to IoT security. It examines the challenges posed by quantum computing and explores PQC techniques, including quantum-resistant cryptographic schemes. A detailed discussion on existing GKM protocols for IoT networks, highlighting their quantum resistance and suitability for secure communication in resource-constrained environments.

2.1 Cryptography Basics

Cryptography is the science behind any cryptographic service involving secure communication over unsecured public channels. Cryptography is primarily used to conceal messages so only intended recipients can read them. Plaintext data is hidden in ciphertext via some encoding method, and only the relevant party can decode the given ciphertext to obtain the original plaintext. Encryption is the process of concealing plaintext in the form of ciphertext, whereas decryption is the process of recovering plaintext from the given ciphertext.

Nowadays, many cryptography-based applications are used in daily life. As the Figure 2.1 describes, cryptography is divided into two branches: Symmetric and Asymmetric (Public-key Cryptography)[27].

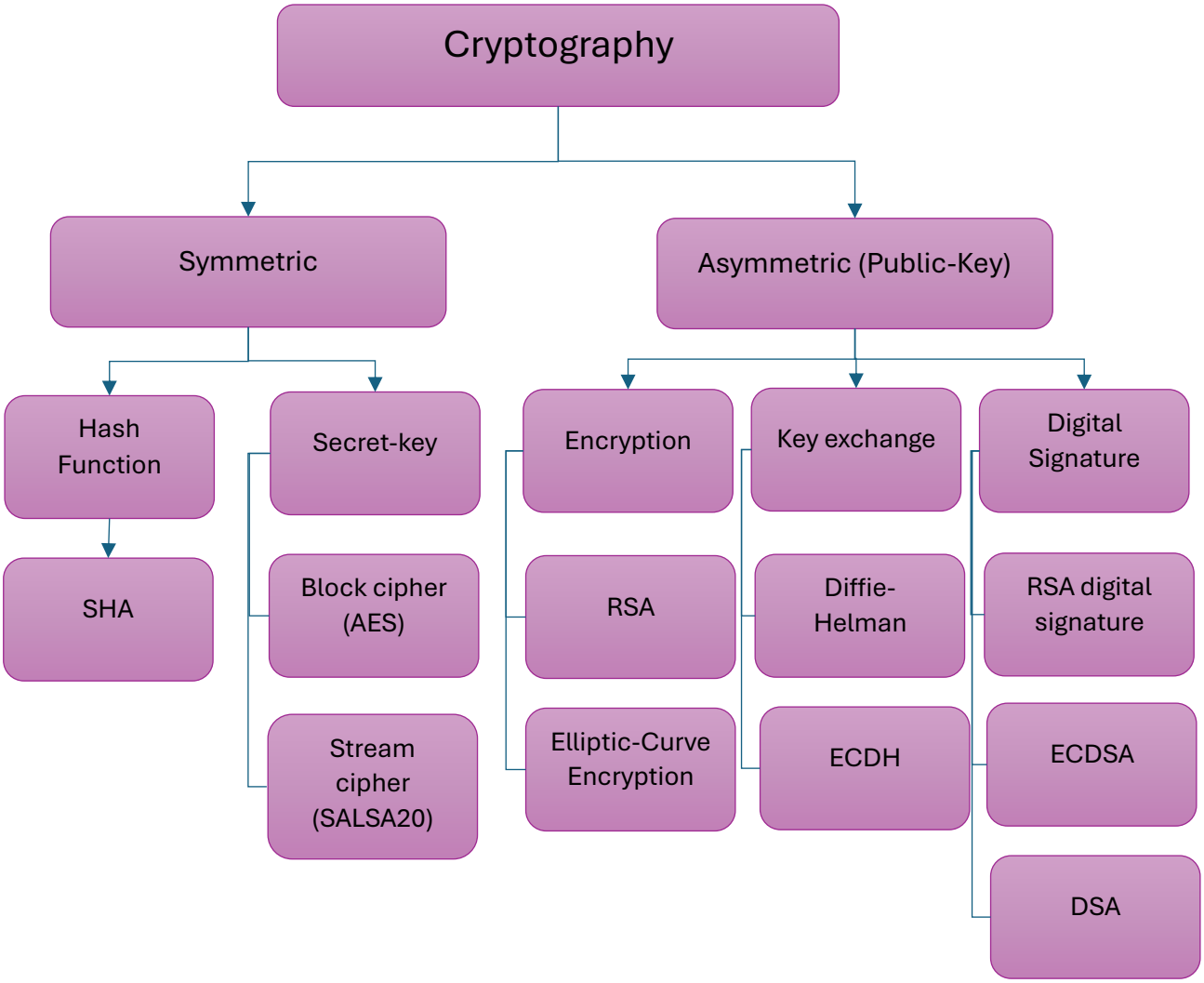

Figure 2. 1 Cryptographic Protocols

2.1.1 Symmetric Cryptography:

In symmetric key cryptography, the same key is used for encryption and decryption, which makes the process faster. The key used to achieve identical levels of security is shorter, relatively safe, and reliable. Because an identical key is used, the key must be exchanged between parties at the beginning of communication[28]. This is an essential step if a significant distance separates both parties. An individual should generate and distribute a new key for each new communication with

a new party. This problem is solved by public-key cryptography. Because two parties share a key, the message's authenticity cannot be guaranteed. Symmetric keys are challenging to maintain.

A block cipher is a method that encrypts a fixed size of n bits of data simultaneously, known as a block. Each block is commonly 64 bits, 128 bits, or 256 bits in size. A 128-bit block cipher, for example, will take 128 bits of plaintext and encrypt it into 128 bits of ciphertext. Most cryptographic schemes in use today are block ciphers. Popular block ciphers are DES, 3DES, AES, Blowfish, and Twofish.

A stream cipher Is a type of pseudorandom bit. To be secure, stream cipher implementation at a time. The key is an infinite stream of pseudorandom bits. To be secure, a stream cipher implementation's pseudorandom number generator should be unpredictable, and the key should never be reused. Stream ciphers are intended to approximate the One-Time Pad. RC4 is one of the most widely used steam ciphers.

2.1.2 Asymmetric Cryptography (Public Key):

In public key cryptography, two keys are used by each participant: a public key, which is available to others, and a private key, which is kept secret. The main requirement of PKC is a trapdoor function. Because of a function's trapdoor property, decryption is only successful if the trapdoor value (the private key corresponding to the public key) matches. In digital signatures, the public key cryptosystem provides both authenticity and non-repudiation. There are several algorithm approaches, such as DLP (Discrete Logarithm Problem), DLP over the elliptic curve (ECDLP), integer factorization, and SVP (Shortest Vector Problem), which are considered computationally hard. The RSA public key cryptosystem uses the integer factorization hardness. RSA is a cryptographic algorithm used in encryption and digital signature algorithms[29]. DLP and ECDLP are used in key exchange algorithms like Diffie–Hellman key exchange and encryption and digital signature algorithms like ElGamal. The RSA algorithm is at the heart of almost all PKC-based

products and standards. In recent years, the size of the RSA key has been steadily increased to provide the required security in response to advancements in software and hardware technologies. This increase in key size necessitated a significant amount of processing for RSA-based applications. This affects services, such as e-commerce sites, that process multiple secure transactions simultaneously. Recent advancements in ECC have created a new alternate primitive with a smaller key size.

2.1.3 IoT Security in Quantum World

It is challenging to provide high security to resource-constraint IoT devices. IoT devices mainly rely on batteries and resource constraints regarding memory and computational power, making it difficult to implement cryptosystems, including extensive computational power and rigorous mathematical operations. And most cyber algorithms rely on insurmountable combinatorial complexity. To protect the IoT node security, the most widely used cryptosystems are symmetric, asymmetric (also known as: - public-key cryptography), and hash functions. The AES is currently the most well-known example of a symmetric cipher, allowing messages to be encrypted with secret keys of lengths of 128, 196, and 256 bits, denoted as AES-128, AES-196, and AES-256, respectively. Among these, AES-128 [16], [19], [30] is the most widely used in IoT security. The most well-known AES attack is a brute-force search of all possible keys. Because Grover's algorithm uses quantum computers to accelerate this process dramatically, the key size of AES must be doubled. That is, an AES key size of 256 bits [1], [18], [31], [32] is required to achieve a security level of 128 bits against quantum computer attacks. Hash functions, public key encryption schemes, signature schemes, and key exchange protocols are the fundamental building blocks of a public key environment. A hash function is a map that converts arbitrary-length data to a hash value of a small, fixed length. As a result, finding two different messages that map to the same hash value should be difficult (collision resistance). Today, the most widely used hash functions

are SHA-2 and SHA-3, members of the NIST-selected Secure Hash Algorithm (SHA) family. SHA-2 can be further subdivided into SHA-256 [1], [13], [19], [20], [31], SHA-384, and SHA512, depending on the output length. According to NIST, we should also increase the output of hash functions to prevent attacks using Grover's algorithm. Public-key cryptography has become essential for Internet communication due to its high security in websites, bank transactions, emails, and healthcare systems to maintain medical data and digital signature documents[33]. TLS (Transport layer security) is used for secure HTTP connections. The public-key cryptography protocols, like RSA, ECC, or DH, are widely used in TLS. Quantum attacks have a far more significant impact on existing public key encryption and digital signature schemes. In [34], [35], [36], a based encryption method is used. In [16], Dashmeet Kaur Ajman et. Al, use RSA method for encryption, signature, and verification process. In [37], polynomials based on RSA modulus are used. In[19] , Nils Gentschen Felde et al. Discuss how cryptography and highly restricted devices exclude one another at first glance. Comparing the cipher requirements with the devices' constraints provides an initial understanding of the contradiction. Because resources – particularly main memory – are limited, some thoughts on limitations and cryptographic choices are introduced. In [1], [14], [20], [30], [31], ECDH-based cryptosystems are used. Their security is based on the hardness of certain number theoretic problems, such as integer factorization and solving (elliptic curve) discrete logarithms. However, Shor's algorithm can efficiently solve these problems on a quantum computer, rendering all traditional schemes insecure as large quantum computers become available.

Table 2. 1 Security effects of a quantum computer

Cryptography protocols	Purpose	Pre-Quantum Security level	Post-Quantum Security level
Symmetric Key			
AES-128	Block Cipher	128	64(Grover)
AES-256	Block Cipher	256	128(Grover)
SALSA20	Stream Cipher	256	128(Grover)
GMAC	MAC	128	128(no impact)
POLY-1305	MAC	128	128(no impact)
SHA-256	Hash Function	256	128(Grover)
SHA-3	Hash Function	256	128(Grover)
Asymmetric Key			
RSA-3072	Encryption	128	Broken (Shor)
RSA-3072	Signature	128	Broken (Shor)
DH-3072	Key Exchange	128	Broken (Shor)
DSA-3072	Signature	128	Broken (Shor)
256-bit ECDH	Key Exchange	128	Broken (Shor)
256-bit ECDSA	Signature	128	Broken (Shor)

To summarize, quantum computers significantly impact the security of all current cryptographic schemes. While it is relatively simple to prevent quantum attacks on symmetric schemes and hash functions (increase the key and output size, respectively), public key schemes such as RSA and ECC are completely broken (Table 2.1).

As a result, we must create new schemes for public key encryption and signatures whose security is based on mathematical problems not affected by quantum computer attacks.

2.2 Post-Quantum Cryptography (PQC):

After identifying the potential of Shor's and Grover's algorithm, researchers began to develop a quantum-resistant algorithm to encounter the threats to current cryptography systems. Post-quantum cryptography (PQC) algorithms are expected to remain stable after functional large-scale quantum computing machines are available. PQC algorithms are typically implemented using lattice-based cryptography, multivariate cryptography, hash-based cryptography, and code-based cryptography (Table: 2.2). The National Institute of Standards and Technology (NIST) published a report on the need for PQC algorithms in 2016 stating that the need for standardizing the new post-quantum cryptosystem had been established for the security of digital communications. Many proposals have been submitted to the National Institute of Standards and Technology (NIST) [38]. NIST received 82 candidate algorithms in 2017, from which 69 were selected as First-Round candidates. 26 candidates are qualified to move forward towards the second Round in 2019. On 22nd July 2020, in the Third Round 7, finalists and 8 alternative candidate algorithms were announced. The third NIST PQC standardization conference will be held in 2022. This article provides an overview of all essential PQCs selected by NIST. This will be a good starting point for researchers to better understand PQC and its future.

2.2.1 The Rise of Post-Quantum Cryptography: Preparing for a Quantum Future

Quantum computing is a revolutionary paradigm that functions based on the principles of quantum mechanics, facilitating computations that exceed the capabilities of classical systems[39]. In contrast to classical computers that utilize bits to denote information as either 0 or 1, quantum computers employ quantum bits or qubits. A qubit can exist in a superposition of states, technically denoted as

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle|$$

Where α and $\beta \in \mathbb{C}$ (complex numbers) are amplitudes satisfying:

$$|\alpha|^2+|\beta|^2=1$$

This superposition enables quantum systems with n bits to represent 2^n states concurrently, providing exponential parallelism.

Quantum computers utilize phenomena like superposition, entanglement, and quantum interference to tackle issues that are computationally impractical for classical systems.

Entanglement, a phenomenon, links the states of several qubits so that the state of one qubit instantaneously determines the state of another, irrespective of distance. An entangled state of two qubits is represented as:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle+|11\rangle)$$

This property is fundamental to several quantum algorithms and is crucial for quantum communication protocols such as quantum key distribution (QKD).

Quantum gates qubits using reversible unitary operations. Common gates include:

Hadamard Gate (H): Creates superposition:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$$

Pauli Gates (X, Y, Z): Rotate qubits:

$$X|0\rangle =|1\rangle, \qquad X|1\rangle =|0\rangle$$

CNOT Gate: Entangles qubits by flipping the target qubit based on the control qubit:

$$CNOT(|10\rangle) =|11\rangle.$$

2.2.1.1 Quantum Algorithms and Their Impact on Cryptographic Security

Quantum algorithms utilize these features to attain substantial computing benefits compared to classical algorithms. Shor's algorithm, for example, effectively factors huge integers by transforming the problem into determining the period r of a function $f(x) = a^x mod\ N$. Shor's algorithm[39] employs the quantum Fourier transform to ascertain r in polynomial time, hence undermining prevalent cryptographic systems such as RSA and ECC. Grover's approach, conversely, offers a quadratic acceleration for unstructured search problems, lowering the time complexity from O(N) to O(√N). This affects symmetric cryptography by reducing the security strength of keys by half, meaning a 256-bit key provides merely 128 bits of security against a quantum adversary[40].

These computational benefits present substantial risks to conventional encryption systems. Shor's algorithm compromises RSA and ECC, which depend on the difficulty of factoring huge integers and resolving discrete logarithms. Although more robust, symmetric encryption techniques such as AES necessitate extended key lengths to mitigate the effects of Grover's algorithm. Hash functions, essential for digital signatures and integrity verification, face decreased security levels due to quantum advancements.

Nonetheless, the development of scalable and functional quantum computers encounters various obstacles. Quantum systems exhibit significant sensitivity to external perturbations, resulting in decoherence, wherein the quantum state collapses prior to the completion of computation. Error correction techniques are crucial for mitigation, yet they necessitate the encoding of a logical qubit into many physical qubits, thereby substantially augmenting resource requirements. Contemporary quantum devices are categorized as noisy intermediate-scale quantum (NISQ) systems, which are constrained in their capacity to execute extensive, error-free calculations.

Furthermore, scaling poses a considerable challenge, as developing dependable quantum computers with thousands of qubits necessitates surmounting substantial technical and materials science obstacles.

The advent of quantum computers has prompted the creation of PQC, which is intended to withstand quantum assaults. Lattice-based cryptographic methods, like CRYSTALS-Kyber and Sabre, exhibit potential owing to their efficiency on resource-limited devices and their dependence on challenging challenges such as MLWE and MLWR. These schemes offer substantial security while ensuring practical performance for real-world applications, rendering them formidable contenders in the NIST post-quantum standardization process.

Quantum computing is a double-edged sword: it possesses significant promise for enhancing areas such as optimization and quantum simulations while concurrently jeopardizing the security of contemporary cryptographic systems. With the advancement of quantum technology, safeguarding critical infrastructures from their potential threats has become imperative, highlighting the necessity of adopting quantum-resistant cryptographic solutions.

2.2.1.2 Quantum-Resistance KEM and Digital Signatures

Quantum-resistant Key Encapsulation Mechanisms (KEMs) and digital signatures are crucial for securing cryptographic systems against quantum computing threats. Conventional encryption techniques, including RSA and ECC, depend on mathematical challenges, such as integer factorization and discrete logarithms, that can be effectively addressed by quantum algorithms, notably Shor's algorithm. This presents a substantial threat to the security of data and communications in the quantum age. As quantum computers advance, it is essential to investigate

encryption techniques that can withstand such threats. KEMs and digital signatures must advance to offer secure alternatives in this environment.

A KEM serves as a cryptographic protocol designed for the secure exchange of keys. KEM enables two parties to securely exchange a symmetric encryption key without direct sharing. The mechanism operates in two phases: encapsulation and decapsulation. During the encapsulation phase, the sender creates a public key and employs it to encrypt a symmetric key, referred to as the encapsulated key, which is subsequently transmitted to the receiver. During the decapsulation phase, the receiver employs their private key to decrypt the encapsulated key, thereby recovering the symmetric key. The recovered key is utilized for the encryption and decryption of substantial data volumes through efficient symmetric encryption algorithms such as AES. This hybrid method utilizes the advantages of public-key encryption for secure key exchange and symmetric encryption for efficient data encryption.

Quantum-resistant key encapsulation mechanisms (KEMs) rely on hard mathematical problems, including those derived from lattices, codes, hashed or multivariate polynomials, which are thought to remain intractable for quantum computers. These constructions guarantee the security of key exchange, even in a quantum environment where conventional schemes would be inadequate. Digital signatures, utilized for data authentication and integrity verification, should likewise rely on quantum-resistant problems. In this subsection, we discuss the NIST-based KEM and digital signature as follows:

- **SABER-KEM**[41]**:** The Saber KEM is a cryptographic scheme based on lattice structures, utilizing the MLWR problem to ensure security. Saber is engineered for post-quantum applications, providing robust resistance to quantum attacks and ensuring computational efficiency. The design prioritizes the reduction of computational and memory overhead,

rendering it particularly appropriate for resource-constrained environments, including IoT devices. Saber establishes a balance between security and performance, offering strong protection without requiring Gaussian sampling in alternative schemes such as Kyber. As a finalist in the NIST post-quantum standardization process, Sabre is a competitive and dependable choice for post-quantum cryptography.

- **CRYSTALS-KYBER**[42]**:** Kyber is a KEM; it is IND CCA-2 secure. Its security is based on the hardness assumption of the LWE problem over module lattices. Kyber is one of the finalists in NIST post-quantum cryptography project Round 3. KYBER's module structure is built on a power-of-two cyclotomic ring, allowing fast computations using the number theoretic transform (NTT). The submission includes three different parameter sets aimed at varying levels of security. José Ignacio Escribano Pablos presents a post-quantum four-round GAKE. They use Kyber encryption scheme as their main building block. As a result, the four-round instantiation can be demonstrated to provide post-quantum security guarantees in the quantum random oracle model under the Module-LWE assumption.
- **CRYSTALS-DILITHIUM**[43]**:** Dilithium is a digital signature scheme based on the hardness of lattice problems over module lattices that is highly secure against specific message attacks. Dilithium has a strong, balanced performance regarding key and signature sizes and the efficiency of the key generation, signing, and verification algorithms. Dilithium performs well in real-world experiments. Dilithium is one of the finalists in NIST post-quantum cryptography project Round 3. N Gupta et al present the most miniature hardware accelerator for CRYSTALS- Dilithium and present a comparison with the existing implementations, which proves that their design is 35% more efficient in the context of area and time [44].
- **FALCON:** FALCON is a signature scheme based on lattices that employs the "hash and sign" paradigm. Security is based on the hardness of the SIS (short integer solution) problem over

NTRU lattices, and proofs of security are provided in the ROM and QROM with tight reductions. Falcon is one of the finalists in NIST post-quantum cryptography project Round 3. Falcon provides security, compactness, speed, scalability, and RAM Economy.

- **SPHINCS+:** SPHINCS+ is a stateless hash-based signature scheme. Its security depends entirely on assumptions about the underlying hash function's security. SPHINCS+ is one of the alternatives in the NIST post-quantum cryptography project Round 3. The reason behind it not considering SPHINCS+ as a finalist in Round 3 is that the trade-off is its speed and size. It would be difficult to imagine it on TLS. So, in Round 2 Status, NIST states that "if NIST sees the need for an additional signature algorithm for applications that need very high security and can tolerate larger and slower signatures, NIST may decide to standardize SPHINCS+ in the future." Later, it includes several enhancements to reduce signature size, making it a winner. Berthet et al. [24] provide an area-efficient FPGA implementation of SPHINCS+. This post-quantum signature scheme ensures high security, allowing its use in embedded systems such as hardware security modules, IoT devices, or nanosatellites.

Table 2. 2 Quantum Resistance Protocols

Cryptograph Algorithm	**PQC family**	**Type**	**Computational problem**	**Provable Security**	**Security type**
CRYSTALS-KYBER	Lattice-Based	Public-Key Encryption/KEMs	MLWR	ROM	CCA security
CRYSTALS-KYBER	Lattice-Based	Public-Key Encryption/KEMs	MLWE	ROM	CCA security
CRYSTALS-DILITHIUM	Lattice-Based	Digital Signatures	MLWE & MSIS	QROM	CoreSVP security
SPHINCS+	Hash-Based	Digital Signatures	Hash Function	ROM	-

FALCON	Code-based	Digital Signatures	SIS	ROM and QROM	low CoreSVP security

Specific algorithms, however, may be too inconvenient to use in IoT networks. Cryptography is a critical technology for securing communication in IoT networks. IoT comprises heterogeneous devices ranging from low to medium power, such as sensors, actuators, edge devices, etc. Dealing with cryptographic techniques in the IoT environment is difficult, as it sometimes necessitates lightweight cryptographic solutions. has launched a multi-year initiative to standardize PQC in anticipation of the imminent quantum menace. Due to their efficiency on constrained devices and their resistance to quantum attacks, lattice-based cryptographic schemes have become viable candidates. In 2022, NIST selected CRYSTALS-KYBER as the new PQC standard, while Saber, another lattice-based KEM, was a finalist. Kyber and Saber are both founded on modular lattice problems. Kyber employs the MLWE problem, while Saber employs the MLWR problem[45]. Saber remains a highly competitive option for post-quantum applications, as it provides comparable security and performance without any deficiencies, despite Kyber's selection as the standard after extensive research on MLWE[46].

To ensure IoT networks can integrate new cryptographic schemes with protocols like Secure Shell (SSH) or Transport Layer Security (TLS). To do so, designers of post-quantum cryptosystems must consider the following characteristics for IoT use cases:

- Transfer delay is caused by encryption and decryption at both ends of the communication line, assuming several devices, from large and fast servers to slow and memory-limited IoT devices.
- Limit the size of public keys and signatures for ultra-low latency.

- A network architecture that enables cryptanalysis and the detection of vulnerabilities in a dense IoT network.
- Integration with the existing infrastructure is flawless.

2.3 Existing Group Key Management Protocols for IoTs

An IoT network's group key management should be efficient and highly scalable. Because of the limitations of IoT devices, any operation performed by the devices should not exhaust the device's resources. Because traditional protocols are insufficient for resource-constrained IoT devices, we require faster and lightweight protocols for secure group communications. As a result, key management in the group should be implemented effectively. The GKM schemes should use the least memory in each device and distribute the group key with the least communication overhead. On the other hand, an IoT network is often dynamic and has many members. To deal with these circumstances, group key management should be highly scalable. Multicast communication reduces terminal bandwidth, energy consumption, and processing overhead. Secure message delivery within a multicast group can be obtained by establishing a group key among the authorized members [14]. The SCG schemes are classified into three categories: centralized, decentralized, and distributed (as shown in Figure 2.2)

Table 2. 3 Notation used in Tables 2.4, 2.7, 2.10

Notations – Description	Notations – Description
N – Total number of users	n- Total number of members in the group.
M – total number of devices	m- Maximum number of sensors
K – key size	k – number of keys
E – Encryption operation	s_k- symmetric key
D – Decryption operation	p – modulus

p_k- public key	t, h- univariate polynomial
g_k- group key	l- number of classes of sensors
	H- hash operation

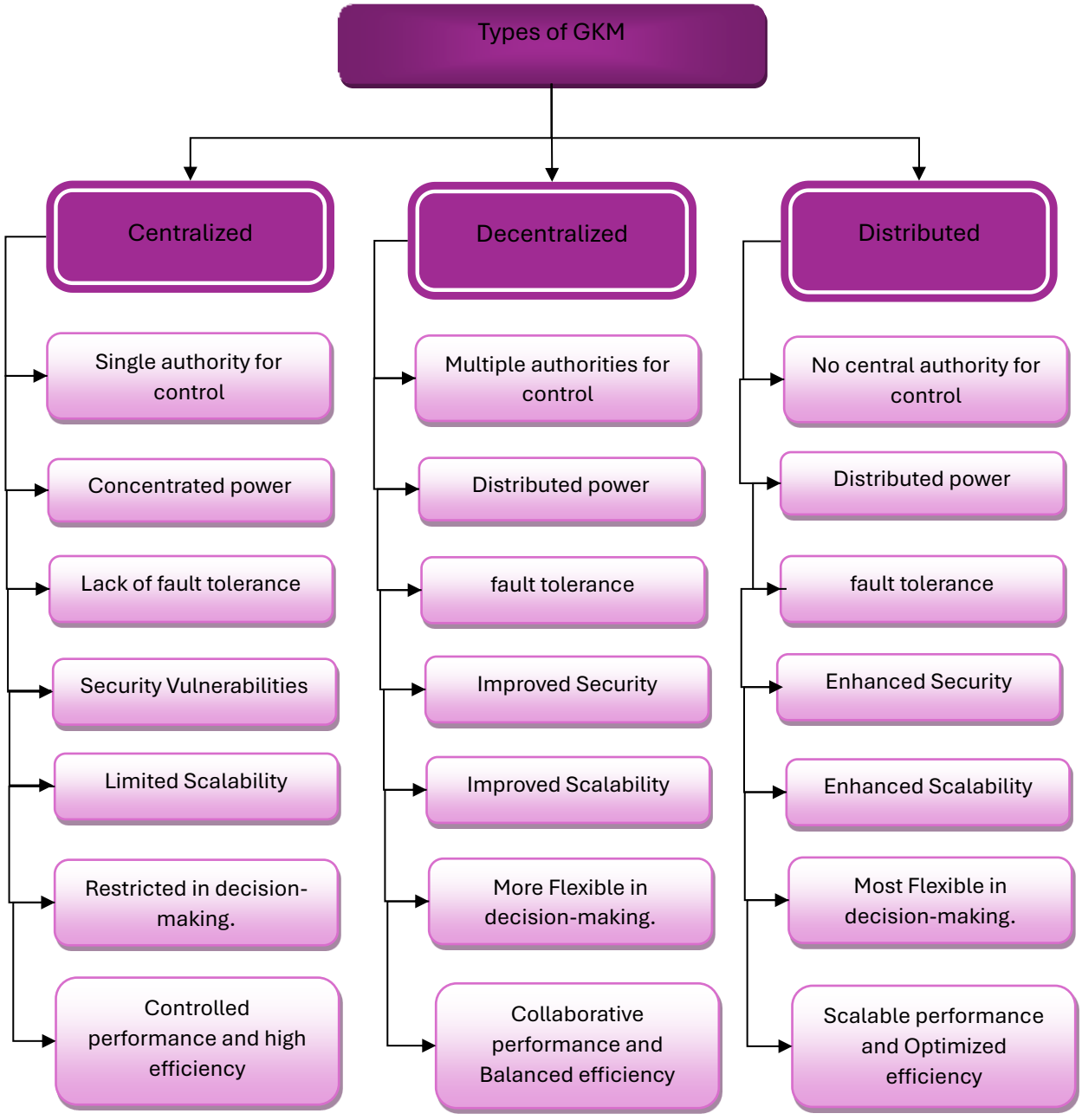

Figure 2. 2 Types of GKM schemes

2.3.1 Centralized GKM schemes:

This section discusses the performance and security of centralized SGC schemes. The supplementary materials provide a more comprehensive explanation of the functionality of the schemes under consideration. The comparison is presented in three tables. The notation used in these tables are detailed in Table 2.3. Tables 2.4 and 2.5 provide a summary of the performance of centralized schemes. Figure 2.3 compares Centralized (GKM) schemes, highlighting the differences in storage, communication, and computation costs. The graph shows the performance characteristics of various GKM schemes and enables a comprehensive evaluation of their viability and effectiveness. The performance characteristics of the given schemes within the given categories are described and compared using asymptotic notation. By assigning low, medium, and high complexities to storage, communication, and computational costs, the notation provides a framework for comparing the scalability and efficiency of each approach. This notation emphasizes the scaling behavior of the costs relative to the input parameters. The asymptotic notation enables a concise and standardized representation of the complexities, thereby facilitating the evaluation and selection of GKM schemes for secure group communication in resource-constrained scenarios within the given category. Tables 2.4 and 2.5 illustrate how various schemes achieve varying performance levels and employ diverse methods. The security aspects of centralized GKM schemes are summarized in Table 2.6. Some schemes are significantly more efficient than others, but they may pose unacceptable security risks to the group in achieving such results. The group is managed by a centralized trusted entity, the Group Controller (GC), in centralized schemes. This includes managing members joining and leaving and renewing the group key. The GC is the only entity controlling all components of an SGC scheme[15]. This centralized approach aims to reduce computational costs and storage requirements for group members[47]. The efficiency of symmetric key encryption and the high security of key selection

and generation are advantages of centralized schemes. However, the GC is a potential bottleneck and a single point of failure. If the GC of a centralized system fails, the system ceases to function entirely. As the only entity responsible for the entire group, the GC is the primary target of centralized system attacks[15].

XKFS is distinguished by its high storage, communication, and computational costs, which scale linearly with the number of network nodes. Thus, it is better suited for fewer nodes and efficient resource management scenarios. However, it may not be suitable for networks with numerous users or limited resources. The CL-EKM scheme is intended to be lightweight and appropriate for dynamic Wireless Sensor Networks (WSN). It facilitates efficient communication for important updates and management when nodes join or leave a cluster, mitigating the effect of compromised nodes. CL-EKM has moderate storage costs but high communication and computation costs, making it more suitable for networks with a moderate number of nodes.

The KMGC plan prioritizes scalability, work efficiency, and reduced communication time. Combining master key and ECC techniques, KMGC reduces the number of keys stored in nodes, increases scalability, and decreases the risk of denial-of-service (DoS) attacks. Additionally, it reduces the energy and time required for crucial negotiations, thereby improving the overall network's effectiveness. However, due to its high communication and computational costs, it may be limited to large-scale deployments. The SBSA scheme provides a group key establishment protocol and a special key management protocol for secure one-to-many communication in hardware-restricted networks. SBSA guarantees security and effectiveness but has high storage, communication, and computational costs. Even with a comparatively large number of users, it is ideally suited for secure communication. LGKMCP scheme efficiently manages offline users and variations in group membership. While it incurs high storage costs, its communication and

computation costs are low. LGKMCP balances storage costs and group administration effectiveness, making it suitable for situations requiring effective group management.

In summary, selecting a centralized GKM scheme includes considering storage, communication, and computational costs and the scheme's suitability for resource-constrained environments and many users. XKFS is better suited for scenarios with fewer nodes, whereas CL-EKM, KMGC, SBSA, and LGKMCP can accommodate more members with variable cost and efficiency trade-offs. Researchers should evaluate these schemes based on their network requirements, considering scalability, efficiency, security, and resource constraints to choose the most appropriate GKM scheme.

Table 2. 4 Centralized GKM schemes (part 1)

GKM Schemes	**Storage Cost**	**Communication Cost**	**Computational Cost**
XKFS[48]	High: $O(n)$	High: $O(n)$	High: $O(n)$
CL-EKM [20]	Medium: p_k/g_k	High: $O(n)$	High: $O(n)$
KMGC[49]	Medium: p_k/g_k	High: $O(n)$	High: $O(n)$
SBSA [50]	High: $O(n)$	High: $O(n)$	High: $O(n)$
G-IKEv2 [19]	Low: $O(1)$	Medium: E / D	Medium: g_k
LGKMCP [51]	High: $O(n)$	Low: $O(1)$	Low: $O(1)$

Table 2. 5 Centralized GKM schemes (part 2)

GKM Schemes	Cryptography type	Key update frequency	Discussion	
XKFS[48]	Hash, XOR, Symmetric	At membership change	Pros: The scheme minimizes key freshness operations and energy consumption.	Cons: the network is vulnerable if the head node fails.
CL-EKM [20]	Asymmetric	At membership change	Pros: Supports efficient key revocation for compromised nodes.	Cons: vulnerable to man-in-the-middle attacks if the nodes do not authenticate each other.
KMGC[49]	Asymmetric	At membership change	Pros: reduces the number of keys stored in nodes.	Cons: Due to the specific requirements for its implementation, the scheme may not be suitable for all types of wireless sensor networks.
SBSA [50]	PRG, symmetric	At membership change	Pros: It does not require special hardware or employ costly cryptographic operations, making it suitable for hardware-constrained networks.	Cons: Key management is required for the broadcast channel, and each node must update the session-specific private key.
G-IKEv2 [19]	AES-128 SHA-256 ECC-256	Periodic	Pros: It is a secure and scalable protocol that can support multiple cryptographic functions, making it highly adaptable to different applications.	Cons: It is still in the early stages of development, so some security issues may need to be addressed before it is ready for widespread use.
LGKMCP [51]	Symmetric, XOR	Periodic	Compared with pre-existing schemes, the proposed scheme has demonstrated efficiency in effectively managing offline users.	

Table 2. 6 Comparison of Centralized GKM scheme regarding security features.

GKM Schemes	Forward / Backward Secrecy	Anti-Collision	Message Confidentiality	Member Authentication	Message Integrity	Group Independence	Instant Rekey	Quantum Resistance
XKFS[48]	✓/✓	X	✓	✓	✓	✓	✓	X
CL-EKM [20]	✓/✓	✓	✓	✓	✓	✓	✓	X
KMGC[49]	✓/✓	X	✓	✓	✓	✓	✓	X
SBSA [50]	✓/✓	✓	✓	✓	✓	✓	✓	X
G-IKEv2 [19]	✓/✓	X	✓	✓	✓	✓	✓	X
LGKMCP [51]	✓/✓	✓	✓	✓	✓	X	✓	X

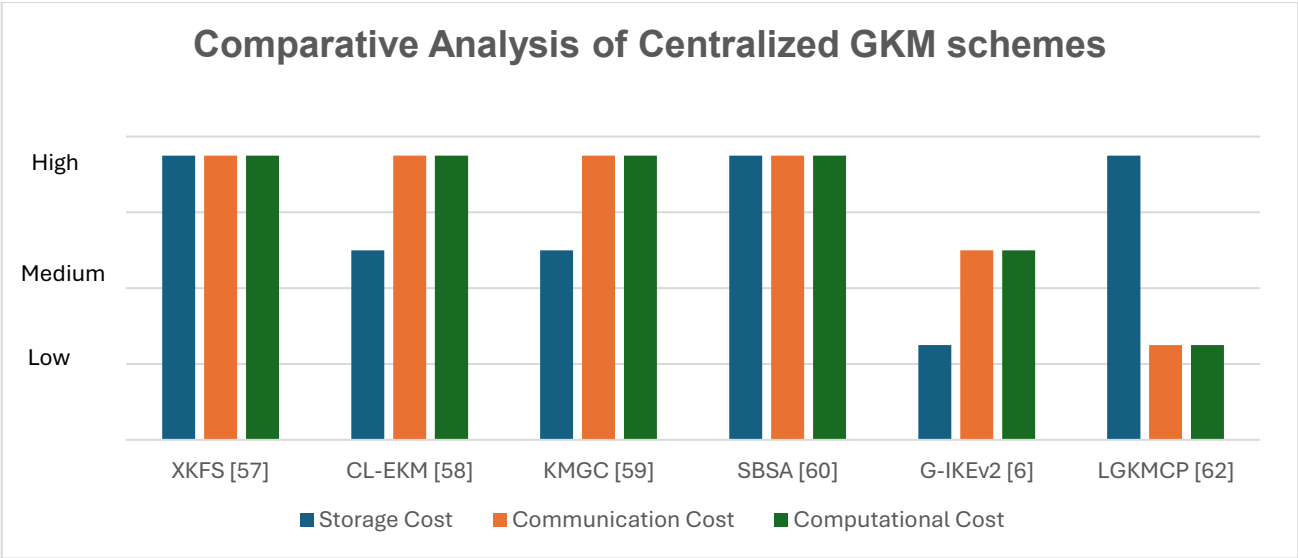

Figure 2. 3 Comparative analysis of Centralized GKM schemes

2.3.2 Decentralized GKM schemes:

In this section, we discuss the efficiency and safety of decentralized SGC schemes. The supplementary material provides a more comprehensive explanation of the functionality of the schemes under consideration. Table 2.3 describes the notation employed in these tables. Tables 2.7 and 2.8 illustrate how various schemes achieve performance levels and employ various techniques. Table 2.9 outlines the security features of decentralized GKM schemes. Figure 2.4 compares decentralized (GKM) schemes, highlighting the differences in storage, communication, and computation costs.

In decentralized architectures, a central unit performs some tasks while others require collaboration. These decentralized protocols aim for efficiency and fault tolerance[15]. The division of group management among SGC is a common approach in decentralized schemes. SGC schemes aim to reduce the problem of concentrating all workloads on a single entity [47]. Another approach is to allocate group key generation to a group controller while all group members do group key distribution collaboratively [15].

DBGK and DLGKM-AC are decentralized GKM schemes for IoT environments with limited resources. They manage group membership efficiently and ensure secure multicast communication. DBGK reduces the rekeying burden caused by dynamic and mobile group membership while preserving both backward and forward secrecy. DLGKM-AC reduces the Key Distribution Center's (KDC) rekeying burden and offers a scalable IoT architecture that improves overall efficiency.

ABP-MAGKE and LT-SMM provide lightweight and efficient protocols for secure group communication in WSNs with limited resources. ABP-MAGKE concentrates on membership authentication and pairwise shared key distribution, whereas LT-SMM deals with frequent membership changes. Both protocols have minimal communication and computational costs,

which reduces the rekeying burden in WSNs. SCBA is a decentralized GKM scheme explicitly designed for WBANs that ensures secure and efficient communication in medical environments with continuous physiological monitoring. SCBA addresses the need for minimal communication and computational costs, but its specific approach to rekeying is not specified.

Table 2. 7 Decentralized GKM schemes (part 1)

GKM Schemes	Storage Cost	Communication Cost	Computational Cost
DBGK [52]	Medium: $O(k * K)$	Low: $O(logk)$	Low: $O(logk)$
DLGKM-AC [1]	Device: Low: $O(log_2 n * s_k)$ User: Medium: $O(m * s_k)$	Low: $O(logk)$	Low: $O(logk + M)$
ABP-MAGKE [12]	Low: $(t + h)log_2 p$	Low: $compute\ n\ hash\ outputs.$ $(2 <= n < N)$	Low: $2h - 1$
SCBA[53]	Smartphone: $O(M + m)$ Sensor: $O(m)$	Low: $O(M)$	Smartphone: $O(M)$ Sensor: $O(1)$
DCSGS[36]	High: $O(N * M)$	High: $O(N * M)$	High: $O(N * M)$
LT-SMM [54]	High: $O(n)$	High: $O(n)$	High: $O(n)$

Table 2. 8 Decentralized GKM schemes (part 2)

GKM Schemes	Cryptography type	Key update frequency	Discussion	
DBGK [52]	Hash	On-demand	Pros: Rekeying does not rely on the estimated departure time of objects. Instead, a mechanism based on demand is utilized.	Cons: 1-affects-n problem because rekeying only affects active members with valid tickets in each area.

DLGKM-AC [1]	ECC, AES	Membership change	Pros: It supports a scalable IoT architecture, which reduces the load caused by rekeying at the core network, reducing the single point of failure.	Cons: It is inappropriate for applications requiring a high level of security.
ABP-MAGKE [12]	Hash, XOR, Symmetric	Session wise	noninteractive, can speed up the communication process significantly.	
SCBA[53]	ECC	Membership change	Pros: It provides secure authentication and GKM for WBANs and is resistant to various attacks.	Cons: Limited by the smartphone's computational power and storage capacity, which may be an issue for some applications.
DCSGS[36]	Asymmetric (roughly same as RSA)	Periodic	When the number of servers (n) in a multi-server environment increase, it is discovered that this scheme is the most efficient when compared to the others.	
LT-SMM [54]	Hash	Membership change	Pros: Offers a secure mobility management scheme that enables secure group communication and efficient group deployment.	Cons: Maintaining the logical tree structure requires a significant amount of computational power and memory, which can be quite costly.

Table 2. 9 Comparison of Decentralized GKM scheme in terms of security feature.

GKM Schemes	Forward/ Backward Secrecy	Anti-Collision	Message Confidentiality	Message Authentication	Message Integrity	Group Independence	Instant Rekey	Quantum Resistance
DBGK [52]	✓/✓	✓	✓	✓	✓	✓	✓	X
DLGKM-AC [1]	✓/✓	✓	✓	✓	✓	✓	✓	X
ABP-MAGKE [12]	✓/✓	✓	✓	✓	✓	✓	✓	X
SCBA[53]	✓/✓	✓	✓	✓	✓	✓	✓	X
DCSGS[36]	✓/✓	✓	✓	✓	✓	✓	✓	X
LT-SMM [54]	✓/✓	✓	✓	✓	✓	✓	✓	X

DCSGS is a decentralized GKM scheme for large-scale networks requiring secure group communication. It may incur additional storage, communication, and computational costs. In contrast to other schemes, it does not expressly address rekeying overhead.

When evaluating these schemes, researchers must consider requirements such as resource limitations, dynamic group membership, secure communication, and rekeying overhead. DBGK and DLGKM-AC are well-suited for resource-constrained IoT environments, whereas ABP-MAGKE and LT-SMM are well-suited for resource-constrained WSNs. SCBA addresses the needs of WBANs, whereas DCSGS focuses on large-scale networks that necessitate secure group communication.

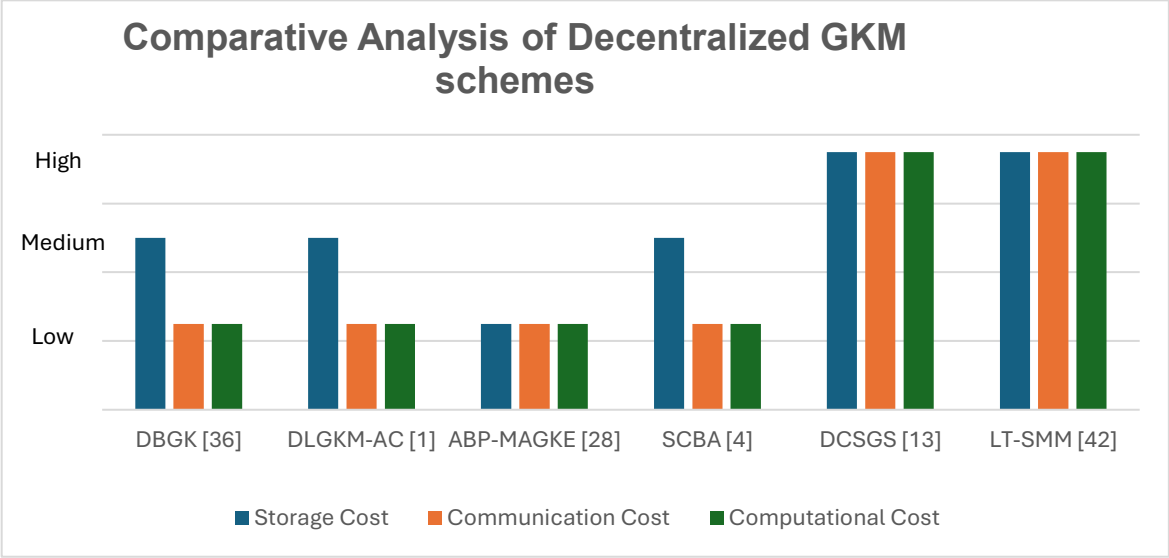

Figure 2. 4 Comparative analysis of Decentralized GKM schemes

2.3.3 Distributed GKM schemes:

In this section, we discuss the efficiency and safety of distributed SGC schemes. The supplementary material provides a more comprehensive explanation of the functionality of the schemes under consideration. Table 2.3 describes the notation employed in these tables. Tables 2.10 and 2.11 illustrate how various schemes achieve performance levels and employ various methods. Table 2.12 provides a summary of the security features of distributed GKM schemes. Figure 2.5 compares distributed (GKM) schemes, highlighting the differences in storage, communication, and computation costs.

In distributed SGC schemes, group members collaborate to manage the group without the assistance of a central authority. Distributed schemes benefit from fault tolerance because no single entity is responsible for distributing and generating keys. However, this comes with increased computational costs for group members and other drawbacks, such as increased device energy consumption.

FMPMAKE, GKPS, and GKAT provide solutions for WSNs with limited resources. FMPMAKE emphasizes efficient membership authentication and key establishment, whereas GKPS

emphasizes efficient key distribution and secure communication. GKAT, on the other hand, focuses on securing group communication in WSNs with mobile decline, thereby providing increased resistance to node capture attacks. GKAT stands out among these schemes when mobile sinks are present, as it addresses the unique challenges posed by node capture attacks.

In dynamic IoT environments, GROUPIT, GKMSFC, and SGKES offer solutions. GROUPIT accommodates varying memberships and device counts, efficiently managing key updates and ensuring secure communication with IoT devices with limited resources. GKMSFC reduces communication costs and message overhead for fog computing networks. SGKES prioritizes the establishment of secure group keys in IoT environments with heterogeneous devices and dynamic group memberships. GKMSFC stands out in comparison due to its scalability, decreased communication costs, and optimized message overhead. SGKES, on the other hand, offers a specialized solution for IoT environments with heterogeneous devices and dynamic group memberships.

ITSKM, MIPUF, GKMCA, and GKA stand out regarding specific IoT applications. ITSKM addresses group-based communication in low-constrained IoT device-to-device (D2D) networks, specifically in medical assisted living scenarios. MIPUF emphasizes key management in energy-efficient IoT devices. GKMCA introduces a group key management scheme for clustered IoT environments, minimizing computational overhead and communication expenses. GKA emphasizes group key agreement with forward secrecy for the distributed Internet of Things environment. ITSKM stands out among these protocols because it incorporates physical layer key exchange, providing an additional security layer against unauthorized access.

Table 2. 10 Distributed GKM schemes (part 1)

GKM Schemes	Storage Cost	Communication Cost	Computational Cost

FMPMAKE[37]	High: $O(n)$	High: $O(n)$	High: $O(n)$
GROUPIT[31]	Medium: $O(log_2 m * k)$	Medium: $O(logm * k)$	Medium: $O(E * D * k * log_2(m * k))$
GKA [55]	High: $O(n)$	High: $O(n)$	High: $O(n)$
GKMSFC [56]	Low: $O(k)$	High: $O(n)$	High: $O(n)$
ITSKM [57]	Low: $O(M * K)$	Low: $O(K * k)$	Low: $O(M)$
GKMCA [58]	Low: $O(1)$	Low: $O(K)$	Low: $O(K)$
GKPS[4]	Medium: $(l\ -\ 1)\ t$	Low: $(l\ -\ 1)$	Medium: $(l-1)(t-1)$ multiplications in Z_N
MIPUF [21]	Low: $O(1)$	High: $O(NlogN)$	High: $O(N)$
SGKES [59]	High: $O(nK)$	High: $O(n)$	High: $O(n)$
GKAT [60]	Medium: $O(E + D)$	Low: $O(H)$	Low: $O(p + H)$

Table 2. 11 Distributed GKM schemes (part 2)

GKM Schemes	Crypto type	Key update frequency	Discussion	
FMPMAKE[37]	Hash	Membership change	Pros: The scheme is secure, with the group key being safe even if m sensors are captured.	Cons: Malicious users may attempt to manipulate the authentication responses or the group key, making the scheme vulnerable. The scheme should be used with other security measures, such as encryption and access control, to reduce these risks.
GROUPIT[31]	AES-256, ECC-224, SHA-256	Membership change	Pros: Our method is more scalable to the expanding IoT environment in terms of both growth rate and overhead.	Cons: Don't support subscriber independence. No scalability, no forward secrecy
GKA [55]	attribute-based encryption	Periodic	Pros: It ensures that each terminal in the group has a unique key and the same set of attributes, preventing unauthorized access.	Cons: Because each terminal must generate and exchange its own key, GKA can be computationally expensive.

GKMSFC [56]	ECC	Periodic	Pros: Reduce cost, rekeying message overhead, and the dependence on reliable channels.	Cons: Difficult to implement due to the bilinear pair calculations, which include scalar multiplications, exponential modules, and pairing calculations.
ITSKM [57]	Symmetric, Asymmetric	Periodic	Pros: Computational complexity is minimized by combining the physical layer key exchange technique (PLKE) with the cryptographic secret sharing approach.	Cons: It is inappropriate for high-constrained IoT D2D communication because the protocol was designed for low-constrained IoT D2D communication.
GKMCA [58]	Symmetric, Asymmetric	Periodic	Pros: Provides control over load balancing among key servers and overcomes key server failover.	Cons: It necessitates a secure and dependable communication channel between the context-aware security server and the key server cluster.
GKPS[4]	Polynomial, RSA	Probabilistic	Pros: One distinguishing feature of proposed GKPS is that it significantly improves the security of polynomial-based schemes.	Cons: Although the scheme provides probabilistic k-secure security, the security can still be compromised after capturing a certain number of sensors.
MIPUF [21]	Hash, AES	Membership change	The scheme's security and overhead analysis show that it is not only secure against multiple attack methods but also low power.	
SGKES [59]	XOR, symmetric	Membership change	Pros: Fast communication due to the use of symmetric keys.	Cons: For some applications, the frequency of key updates may be excessive. - The possibility of denial-of-service (DOS) and reply attack vulnerabilities.
GKAT [60]	Diffie–Hellman problem	Membership change	The results of the performance analysis indicate that the proposed scheme is more efficient in terms of computational complexity and computational time than the literature that has been referenced.	

Table 2. 12 Comparison of Distributed GKM scheme in terms of security features.

GKM Schemes	Forward/ Backward Secrecy	Anti-Collision	Message Confidentiality	Member Authentication	Message Integrity	Group Independence	Instant Rekey	Quantum Resistance
FMPMAKE[37]	✓/✓	✓	✓	✓	✓	✓	✓	X
GROUPIT[31]	✓/✓	✓	✓	✓	✓	✓	✓	X
GKA [55]	✓/✓	✓	✓	✓	✓	✓	✓	X
GKMSFC [56]	✓/✓	✓	✓	✓	✓	✓	✓	X
ITSKM [57]	✓/✓	✓	✓	✓	✓	✓	✓	X
GKMCA [58]	✓/✓	✓	✓	✓	✓	✓	✓	X
GKPS[4]	✓/✓	✓	✓	✓	✓	✓	✓	X
MIPUF [21]	✓/✓	✓	✓	✓	✓	✓	✓	X
SGKES [59]	✓/✓	✓	✓	✓	✓	✓	✓	X
GKAT [60]	x/x	✓	✓	✓	✓	✓	X	X

Comparing the provided GKM schemes reveals that each scheme is tailored to specific environments and applications. GKAT offers enhanced resilience to node capture attacks in scenarios with mobile sinks for resource constrained WSNs. GKMSFC excels in scalability, reduced communication costs, and optimized message overhead in dynamic IoT environments, whereas SGKES provides a

specialized solution for heterogeneous devices and dynamic group memberships. ITSKM's incorporation of physical layer key exchange enhances the security and robustness of specific IoT applications. These comparisons emphasize the unique contributions of each scheme to group key management, considering their respective environments' strengths and benefits.

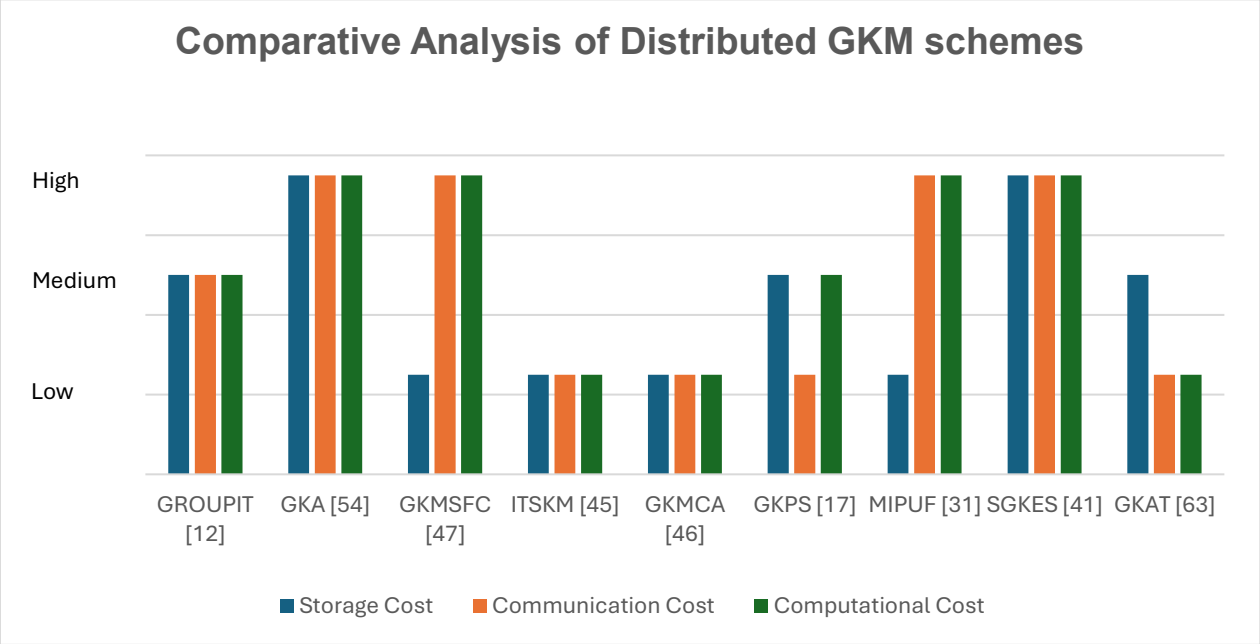

Figure 2. 5 Comparative analysis of Distributed GKM schemes

2.3.4 Security Analysis of GKM Schemes:

In section 1.1.3, "Secure Group Communication," we defined the security requirements for secure GKM schemes. This section presents how the selected secure GKM schemes fulfil these requirements. Tables 2.6, 2.9, and 2.12 compare different centralized, decentralized, and distributed GKM schemes regarding 8 different security factors.

DBGK [52], DCSGS [36]achieve anti-collision by using a unique identifier for each member of the group, which allows messages sent from different members to be distinguishable and identifiable, and also guarantees forward and backward secrecy by preventing collusion attacks from unauthorized

users or devices. DLGKM-AC [1] employs a hierarchical architecture comprised of one Key Distribution Centre (KDC) and several Sub Key Distribution Centers (SKDCs), while GROUPIT[31], uses a device grouping technique and MIPUF [21] using a novel physically unclonable function (PUF) that allows each device to encrypt its data with a unique key though preventing collisions between devices in different groups. ABP-MAGKE [29], FMPMAKE[37] , and GKPS [4] did a polynomial calculation to authenticate memberships and establish a secret session key among all communication entities. SCBA employs a certificateless biometric authentication process to achieve anti-collision, message confidentiality, member authentication, message integrity, and group independence. This entails using representative features from electrocardiogram (ECG) records as distinct biometric parameters during the authentication procedure, allowing for efficient identification of participating sensors without collisions.

Next, GKAT [60] includes ciphertext retention, hidden attribute authentication, and multi-policy access. In the proposed edge-cloud collaborative network architecture, IoT terminals use a key algorithm to produce public and private keys. The cloud server also verifies the terminals' private and public keys. Encrypting IoT terminal cryptographic attributes allows the cloud to authenticate and grant rights for each attribute. The terminal's permissions encrypt FL model parameters and send them to the edge server as sensitive data. The edge server stores the ciphertext's decryption parameters for different FL terminal variants.

By segmenting the shared secret key, GKMSFC [56] achieves anti-collision, message confidentiality, member authentication, message integrity, and group independence. Each segment is divided into two factors, each with its production mechanism, allowing for quick key updating when a new user joins or leaves the fog node. On the other hand, SBSA [50] achieves all security factors using a special key management mechanism. This involves each node having unique session-specific private keys for every broadcast communication session, which helps prevent collisions between different nodes in the

network and provides secure encryption when sending messages over public networks without compromising user privacy or security. LGKMCP [51] successfully attains both forward and backward secrecy, guaranteeing message confidentiality. Employing a unique secret key for each user can effectively guarantee member authentication and message integrity. However, LGKMCP is a centralized key management scheme that uses the key distribution Centre (KDC) to generate the group key. Consequently, it has no group independence. LGKMCP exhibits a consistent expense for rekeying and upholds a publicly accessible bulletin board to enable instant rekeying.

Further, different schemes maintained their message confidentiality by different techniques in DLGKM-AC use an efficient key updating process where all keys are completely independent of each other to safeguard data security, while GKPS, GROUPIT, DCSGS, DBGK [52] encryption of all data sent over the network to ensure its privacy. To provide message confidentiality and integrity, ABP-MAGKE [12], FMPMAKE uses a user authentication process, which allows only authorized members of the group to access messages sent within it. Member Authentication ensures that only legitimate members can join a group communication session, and message integrity guarantees that no malicious third party has tampered with any exchanged data or messages in transit between two parties.

However, message integrity guarantees that any changes to transmitted information will be detected so they cannot go unnoticed. DBGK [52] used a ticket-based system. This entails creating tickets sent to each new joining node, which must be validated before being accepted into the network. Furthermore, our protocol relies on an Area Key Management Server (AKMS), which oversees authenticating members within its corresponding area to ensure that only valid nodes can join or leave groups without compromising security and privacy requirements. DLGKM-AC [1], [52] Member authentication is done via the master token management protocol, ensuring only authorized users can access the system. Message integrity property is ensured with cryptographic techniques such as digital signatures or message authentication codes for verifying messages sent between devices/users within

a group communication session. DCSGS Member Authentication verifies each group member before allowing them access, and Message Integrity checks for errors or tampering with transmitted information.

Next, Group independence allows for multiple groups with overlapping membership without compromising security. It is achieved in DBGK[52] by generating Traffic Encryption Keys with a one-way function (TEKs). This ensures that the data used as an input cannot be retrieved from the resulting output, implying that disclosing one key does not provide an attacker with any additional information needed to retrieve previous, future, or other keys.

Furthermore, invalidating received tickets when a new member joins or existing members leave helps to ensure the security and privacy of all members within each group, DLGKM-AC [1] employs a hierarchical structure comprising a single Key Distribution Centre (KDC) and multiple Sub Key Distribution Centers (SKDCs). During the group communication sessions, the SKDCs oversee the administration of the keys linked to each user or device. The division of key management responsibilities among multiple entities prevents any one entity from possessing all session keys, thereby enhancing security measures against unauthorized access to ongoing interactions. FMPMAKE [37], ABP-MAGKE [12] employing k-secure key confidentiality. This means secure transmissions are possible even in groups of varying sizes, allowing multiple users to join and leave the group without interfering with one another's communications.

Furthermore, Instant Rekey allows users to quickly update their encryption key when needed without having to wait for all other participants in conversation first. DBGK [52] a hierarchical protocol, such as the Logical Key Hierarchy (LKH) or the One-way Function Tree Protocol, which improves on LKH. This approach reduces the number of messages exchanged at the expense of a high computational cost, allowing for quick key updates when members join/leave events or mobility issues within dynamic networks require it. In DCSGS [36] and ABP-MAGKE [12] a distributed key

generation protocol enables group members to generate new keys quickly and securely without waiting for an external source or administrator, allowing for instant data rekeying in the event of a security breach.

Moreover, all SGC schemes listed in Tables 2.6, 2.9, and 2.12 employ symmetric and asymmetric cryptography, which are susceptible to quantum attacks. Symmetric cryptography is influenced by quantum computing. Quantum computers can break AES and 3DES. Symmetric cryptography is secure if the key size is raised. For instance, increasing AES's key size from 128 bits to 256 bits can make it safer against quantum attacks. However, it is essential to observe that this is only a temporary solution, and that post-quantum cryptography should be considered for long-term security [25].

Finally, forward and backward secrecy properties are guaranteed by preventing collusion attacks against unauthorized users trying to access ongoing communications. SCBA[53] Instant Rekey & Forward/Backward Secrecy properties are enabled through Ciphertext Policy Attribute Based Encryption (CP-ABE) technology, which provides dynamic updates when needed, ensuring security remains intact even after keys become compromised due to revocation, etc. DBGK ensures forward secrecy by invalidating received tickets when a new member joins or an existing one leaves the group. In contrast, backward secrecy prevents joining members from accessing communications that occurred before they arrived in the group, and mobile members cannot decrypt stored messages encrypted with previous traffic encryption keys upon movement from one area to another. GKMSFC ensures both forward and backward security by dividing the shared secret key into segments, which are then split into two factors with their production mechanism, allowing quick updating when new users join/leave without the need for additional messages from end-users requesting rekeying operations, thereby significantly reducing cost and network load.

Due to the impending quantum computing revolution, the security of classical cryptographic protocols is at risk. To address this challenge, researchers have been investigating new GKM protocols with

quantum resistance capabilities. This section contrasts and analyses these protocols to determine their suitability for IoT network deployment.

2.4 Group Key Management Applications/Usage Areas:

The applications and usage areas of these applications in the context of SGC schemes are discussed in this section. In recent years, the IoT has gained appeal among end consumers due to its ubiquitous use and range of applications. Applications of the Internet of Things can be found virtually everywhere, including in industrial control, smart healthcare, smart grid, transportation systems, and logistics [63]. IoT is a self-configuring, intelligent system that can connect to a variety of technologies, such as cloud computing, fog computing, radio frequency identification (RFID), and wireless sensor networks (WSN), to share sensory data and control objects with or without human intervention. Due to the inherent promise of this technology, it has already experienced exponential growth in a vast array of use cases across numerous application areas. As experts from across the world continue to examine its capabilities, there is universal consensus that for IoT to reach its full potential, a network architecture that supports security, privacy, and trust must be implemented.

2.4.1 Intelligent Transportation System:

An intelligent transportation system (ITS) is a collection of advanced technologies, such as connected vehicles, cloud computing, and the IoT, used to enhance the safety and efficiency of transport networks. ITS systems collect data about traffic conditions using sensors, cameras, and other devices, which computers or algorithms can then analyse for improved decision-making. This reduces traffic congestion on roads and improves road safety by increasing visibility of potential hazards such as accidents and bad weather. In addition, these systems assist in optimizing fuel efficiency by providing real-time information regarding optimal routes based on current traffic conditions. As modern vehicles

and communication technologies advanced rapidly, people began to believe that the Intelligent Transportation System (ITS) would be implemented within a decade.

ITS integrates information technology into transportation infrastructure to enhance road safety and traffic flow. Nonetheless, security remains a primary concern for vehicular communication systems (VCSs). With a secure group broadcast, this issue can be resolved. Therefore, secure key management schemes are an indispensable network security measure. CAN [61], VANETs and Blockchain technology [62] plays an important role in ITS.

One essential component is the secure distribution and management of cryptographic keys for group communications. For instance, Controller Area Networks (CANs) within vehicles facilitate data exchange among Electronic Control Units (ECUs) and require efficient group key exchange protocols to protect multicast messages and ensure real-time performance. Similarly, Vehicles Ad-hoc Networks (VANETs) enable dynamic communication among mobile nodes via a trusted authority that distributes group keys; however, the dynamic nature of these networks poses challenges in maintaining up-to-date keys as nodes join or leave. In addition, blockchain technology has been proposed as a solution to address vulnerabilities related to centralized key management by offering a distributed, secure mechanism for the storage and transmission of encryption keys.

Despite these advances, unresolved issues remain—especially regarding the efficient, secure management of keys in heterogeneous and dynamically changing vehicular environments. Future research must continue to develop novel cryptographic frameworks and key management solutions tailored to the unique requirements of ITS.

2.4.2 E-Healthcare Management Systems

The integration of IoT technologies in e-healthcare management systems has significantly expanded opportunities for remote monitoring, patient data collection, and secure device communications. Critical components such as Wireless Body Area Networks (WBANs) and Wireless Mobile Environments (WMEs) play pivotal roles in facilitating real-time health monitoring and efficient healthcare delivery.

WBANs employ wearable or implantable sensors to collect vital physiological data, which is transmitted to healthcare centers (HCs) via personal controllers (PCs). Group Key Management (GKM) ensures secure communication between these devices by managing the distribution and update of encryption keys. Traditional group key establishment protocols, however, face challenges in IoT-based E-HCS due to computational overhead and scalability limitations. To address these issues, methods such as the Chinese Remainder Theorem (CRT) and Coded Cooperative Data Exchange (CCDE) have been proposed, enhancing efficiency and reducing storage requirements. Nevertheless, limitations persist, particularly concerning the restricted power capabilities of WBAN sensors and the need for secure group notifications to diverse patient groups.

WMEs extend secure communication capabilities to broader mobile healthcare scenarios, utilizing wireless networks like cellular and Wi-Fi technologies. Group Key Management Protocols (GKMPs) facilitate secure data exchange in mobile environments, although existing solutions struggle with resource constraints and dynamic group changes. To mitigate these challenges, several frameworks have been proposed, including the Healthcare Key Management (HCKM) system, which aims to minimize rekeying overhead while maintaining forward and backward secrecy. Research highlights the ongoing need for efficient key management, low-latency secure communication, and adaptable

authentication mechanisms, particularly as healthcare services increasingly rely on telemedicine technologies.

Overall, both WBANs and WMEs underscore the critical importance of secure, efficient group key management in e-healthcare systems, especially within resource-constrained IoT environments. Future work must continue to address scalability, dynamic membership changes, and the unique challenges posed by the sensitive and critical nature of healthcare communications.

2.4.3 Smart Grid Management System:

Using information technology, SMART Grids are altering the conventional services offered by existing electrical grid networks. They maximize the use of information technology to achieve system efficiency and dependability. In addition to power generation and transmission utilities, smart grids include appliances, meters, sensing devices, and information gateways that function in near real-time [63].

The key components of smart grid technology are as follows:

• **Supervisory Control and Data Acquisition System (SCADA):** This system monitors and controls electrical flow, ensuring that electricity is distributed efficiently and reliably. It is an essential component of smart grid networks since it assists in collecting and analyzing real-time data from remote places to optimize industrial processes[64].

• **Advanced Metering Infrastructure (AMI):** This is the system that collects, measures, and analyses energy usage data from networks with smart meters.

• **Communication Networks:** These networks allow bidirectional communication between various grid components, including power plants, substations, and consumers. Depending on bandwidth requirements, they could be optical fibers or Ethernet passive optical networks.

• **Software & Hardware Components:** These include software applications that manage client accounts, billing systems, etc., as well as hardware components such as routers and switches that enable the safe transfer of data across several nodes in the grid.

GKM is required in AMI and SCADA[65] systems of Smart Grids. Key management is one of the most pressing open issues in smart grids [66]. This necessitates the development of a secure and fast mechanism for access verification of many intelligent gateways and terminal devices. In a different research, the author highlights.

- Developing efficient authentication mechanisms for secure communication between various grid components, including SMGWs, consumer consuming/generating devices, etc[63] .
- Designing lightweight security techniques for Smart Grids' wireless sensors with limited resources [63].
- Investigating new approaches, such as PUF-based KMS, that have not been thoroughly addressed in the literature [63].
- Construct a GKM protocol that can handle collusion assaults, in which a newly added member attacks in collaboration with an eliminated member [64].
- Construct a protocol that can handle the dynamic nature of SCADA systems, in which new members can join or leave at any time [64].
- How to protect multicast communications' secrecy, integrity, and authentication while employing publish-subscribe topologies.
- How to efficiently distribute GKs in large clusters with numerous nodes and subgroups, update or revoke keys safely, and maintain security and performance[67] .
- Data privacy and protection against malicious assaults, such as man-in-the-middle and replay attacks, continue to present obstacles [68].

2.4.4 Air Traffic Management:

ATM is the system that controls air traffic in controlled airspace. It monitors and manages aircraft movements, guaranteeing their safe separation while optimizing their flight paths for efficient travel. The International Civil Aviation Organization (ICAO) establishes global standards for ATM systems, with each nation implementing these standards following local needs and legislation[71]. The primary elements of an ATM system are communication, navigation, surveillance technology, and operational procedures that ensure flight safety, such as route planning and conflict resolution tactics.

The main components of ATM are [69]:

- **Communication technologies:**

It allows air traffic controllers and pilots to communicate with one another. This comprises voice radio in addition to data link systems like the LDACS. It is an air/ground communications system that enables the modernization of ATM. It satisfies special requirements for the L-band environment and ATM applications, making it suitable for use in the modernization of air traffic management systems [70].

- **Navigation technologies:**

It provides information regarding an aircraft's position relative to other objects or geographical characteristics. GPS navigation devices and instrument landing system beacons are included for precise airport approaches.

- **Surveillance technologies:**

It enables ground control operators to monitor an aircraft's location relative to its flight plan path or designated airspace boundaries using radar tracking or Automatic Dependent Surveillance-Broadcast technology (ADS–B).

2.4.4.1 L-Band Digital Aeronautical Communication System (LDACS):

Due to the rising amount of air traffic, the current aeronautical communication technologies have reached their limits. To digitalize formerly analog systems and prepare them for future demands, a process of modernization is undergoing [70]. As part of this transition, the LDACS was developed to replace legacy analogue voice communications to provide secure communication channels for critical infrastructures by implementing Mutual Authentication and Key Establishment protocols as well as Group Key Management procedures that permit authorized users within an LDACS cell or network to access data securely.

GKM is essential since it aids in securing LDACS control channel communications. GKM entails the use of cryptographic mechanisms, such as Mutual Authentication and Key Establishment procedures, to safeguard the data being communicated across a network or among a group of users against unauthorized access. By employing these security measures, LDACS can provide robust cybersecurity when deployed in key infrastructures such as the aviation and aeronautics industries. In this study [71], the author investigates GKM techniques for LDACS control channels and how they promote secure communication within these networks. However, the application of security mechanisms such as GKM approaches on a group-by-group basis, which could provide further protection against hostile actors and illegal access attempts, has not yet been studied. In addition, it investigates how Chinese Remainder Theorem-based algorithms can be implemented in an LDACS system while accounting for their higher message size needs [71].

Regarding LDACS control channels, several possible future paths could be studied. These include more profound research on the implementation of GKM procedures and how they can provide enhanced security against malicious actors and unauthorized access attempts. In addition, it would be advantageous to research new cryptographic algorithms that may give better performance than

solutions based on the Chinese Remainder Theorem while still offering appropriate security for these networks [71].

In short, our investigation into multiple areas of application revealed distinct obstacles and prospective remedies pertaining to protecting group communication and managing cryptographic keys. Although specific challenges have been addressed in previous research papers, unresolved aspects remain that offer potential for further research and development.

2.4.5 Security Analysis of Scenario-Based GKM Protocols:

The GKMVCC [72] and GKECAN [61] schemes are designed for ITS (Intelligent Transportation System). LCGKA [66], PACGKA [67], GKML [13] protocols are designed for smart grid networks, GROUPIT [18], DLGKM-AC [8] considers the hotel management scenario, Protocols AFGKM [19] SCBA [20], are design for e-health system scenarios. The security analysis of all the mentioned protocols depends upon the 8 features which had been defined in [11].

By employing a broker to control group membership and prevent collisions between users in the vehicular cloud, GKMVCC [72] achieves anti-collision. While member identification and integrity are provided using digital signatures, message secrecy is achieved using public key encryption. Group independence and instant rekey are achieved by efficiently allowing users to join or leave the group without activating a fully new key agreement mechanism. Although the GKMVCC [72] scheme uses a combination of conventional authenticated group key agreements, public key encryption, and signature to ensure secure communication among users in the vehicular cloud, which can provide some level of forward and backward secrecy. The GKECAN scheme ensures message confidentiality through the utilization of symmetric shared keys, which are distributed to each Electronic Control Unit (ECU) in the group by a Secure Onboard Communication Unit (SoECU). Additionally, member authentication is achieved by employing a master secret key generated by the SoECU. Furthermore,

the scheme maintains group independence by enabling the removal of a member from the group without compromising the security of the remaining members.

The GKML [13] scheme does not explicitly provide anti-collision properties. Still, by reducing the number of messages sent, which can lower the likelihood of collisions, it ensures message confidentiality by encrypting multicast messages using the group key, member authentication by using Shamir's secret sharing scheme, message integrity by using hash chains to renew the group key and guarantee that multicast messages have not been tampered with, and group independence by using hash chains. The DLGKM-AC[1] scheme prevents collusion attacks, ensures forward-backward secrecy, and ensures secure group communication by using independent keys for each group, employs encryption and message authentication codes (MACs) to ensure that messages exchanged between group members are kept confidential and protected from unauthorized access, whereas GROUPIT [31] scheme achieves anti-collision by utilizing a device grouping technique that allows each device to encrypt its data with a unique key and prevents collisions between devices in different groups, utilizes cryptographic algorithms such as AES encryption/ decryption for secure communication, HMACs are used for member authentication. In contrast, SHA256 hash functions are used to verify message integrity.

Furthermore, SCBA [53] provides the desired security qualities and resilience to various threats. Using a unique identity for each sensor and a certificateless public key scheme to achieve anti-collision. Message secrecy is achieved by encrypting messages using a session key shared by all participating sensors. During the authentication operation, the representative features of the acquired electrocardiogram (ECG) recordings are used as the distinguishing biometric parameter. A hash function is used to ensure message integrity. Using a certificateless public key system, which eliminates the necessity for a certificate authority, provides group independence. Immediate rekeying is enabled using a dynamic key update system requiring minor sensor modifications. Generating a

unique session key and the regular rotation of session keys for each communication session ensures forward and backwards secrecy.

Table 2. 13 Comparing Scenario-Based GKM schemes in terms of security features.

No.	GKE Schemes	Forward/Backward Secrecy	Anti-Collision	Message Confidentiality	Member Authentication	Message Integrity	Group Independence	Instant Rekey	Resistance to Quantum Attack
1.	GKMVCC [10]	✓/✓	X	✓	✓	✓	✓	✓	X
2.	GKECAN [15]	X / X	X	✓	✓	X	✓	X	X
3.	LCGKA [12]	✓/✓	✓	✓	✓	✓	✓	✓	X
4.	PACGKA [16]	✓/✓	✓	✓	✓	✓	✓	✓	X
5.	GKML [17]	X / X	✓	✓	✓	✓	✓	✓	X
6.	GROUPIT [18]	✓/✓	✓	✓	✓	✓	✓	✓	X
7.	DLGKM-AC [37]	✓/✓	✓	✓	✓	✓	✓	✓	X
8.	AFGKM [19]	✓/✓	X	✓	✓	✓	X	X	X
9.	SCBA [20]	✓/✓	✓	✓	✓	✓	✓	✓	X

2.5 Post-Quantum Key Management Schemes

A “multi-party key exchange” expands the “two-party key exchange” that establishes group key exchange (GKE). It requires using a two-party key exchange as a fundamental component in constructing a GKE protocol. The GKM protocol permits secure key exchange between multiple participants. The purpose is to establish a shared group key accessible to all parties involved. The most prominent examples of such construction are [73], [74]. The practical, interactive conference key distribution system proposed by Burmester and Desmedt [74] is afterwards analyzed in [75] concerning the security against passive attacks in the scenario where the Diffie-Hellman problem is intractable. A subsequent compiler was defined by the same authors in [74]. This compiler utilized a tree structure to represent users and incorporated three distinct models: provably secure, unconditionally secure, and depends upon the “Diffie-Hellman problem”, which is further improved in [76]. However, all these compilers depend upon the “Diffie-Hellman” problem, which is not a generic model. Later, Just and Vaudeney [77] developed a generic compiler capable of deriving group protocols from any two-party construction without requiring a particular two-party protocol as a foundation. To authenticate single users or verify their legitimacy as group members, it is necessary to implement a public key signature scheme in any of these compilers. This is essential because of the extra criteria for key management and certification. To address this problem, Abdalla et al[78] have proposed a compiler that uses pair-wise distributed credentials for authentication. The method depends upon a compiled “two-party” solution.

Advancements in isogeny-based cryptography have greatly impacted group key management techniques, including crucial contributions by [79], [80], [81], [82]. Takashima [82] presents group key exchange (GKE) protocols that use secure key derivation functions (KDFs) based on static assumptions to enable efficient and secure key setup inside a group. Fujioka [80] suggests

two GKE protocols, n-UM and BC n-DH, on CSIDH (Commutative super-singular isogeny Diffie-Hellman) hard problems that use cryptographic invariant maps (CIMs) on elliptic curves to establish keys even on insecure channels securely. These protocols benefit from reduced communication rounds and compact key sizes but can be computationally intensive due to isogeny operations. Hougaard [79] introduces SIT, a novel GKE protocol based on super-singular isogenies, including constant-round communication and memory complexity of logarithmic order, to improve efficiency compared to earlier methods. Furukawa [81] discusses the importance of quantum-resistant multi-party key exchange and introduces protocols such as GSIDH and SIBD, which are based on the generalization SSDDH (GSSDDH) and super-singular isogeny decisional Diffie–Hellman (SSDDH) assumption. These protocols address security concerns for multi-party settings but may face limitations in scalability as the number of users increases.

On the contrary, certain authors introduce protocols based on Lattice assumptions [73], [83], [84], [85], [86]. Gareth T. Davies[85] presents a cloud-assisted asynchronous key transport protocol that enhances security against quantum attacks using blinded key encapsulation mechanisms (BKEMs) built on challenging lattice problems. This approach benefits from offloading computational overhead to the cloud but introduces trust issues related to infrastructure security. Pablos [86] suggests a four-round GAKE protocol that combines elements from the Kyber suite of post-quantum tools to establish a secure key agreement on unsecured communication channels. He also demonstrates a practical application of the protocol using the Kyber suite, showcasing its effectiveness and efficiency in the QROM [73]. However, its performance heavily depends on Kyber's suitability for diverse real-world scenarios. Apon[84] presents a protocol for unauthenticated GKE that operates in constant rounds. The approach is built on the Ring-LWE problem and utilizes the Katz-Yung compiler with any post-quantum signature method to enhance scalability and ensure post-quantum security. Choi[83] introduces an innovative GKM protocol

based on lattices with dynamic membership. The protocol extends prior work by adapting it to the RLWE environment and offering security guarantees based on RLWE assumptions and the durability of digital signatures. These lattice-based protocols are generally efficient and scalable but require larger key sizes and higher memory than isogeny-based methods.

Instead of constructing a two-party construction, other proposals are constructed generically from KEMs. However, approximately eleven Post-Quantum GKM schemes are proposed in the literature; F. Samiullah et al. [22] evaluated the suitability of these protocols in the context of an IoT network based on security parameters defined in [11] some open problems that need to be addressed in those protocols. Nevertheless, the current research on PQC performance in IoT applications has a narrow focus, often specific algorithms, and lacks consistent performance metrics. Future research must prioritize NIST-approved KEMs and signatures to promote wider adoption.

Our proposed protocol meets the specific needs of IoT networks by utilizing the Module-LWR assumption, which guarantees both post-quantum security and computing efficiency. In contrast to isogeny-based systems (e.g., GSIDH and SIBD), which are computationally intensive and less feasible for resource-limited devices, our lattice-based protocol balances security and efficiency. Furthermore, we eliminate the necessity for expensive signature methods by using the Abdalla et al. compiler utilizing pairwise distributed passwords for authentication. Compared to previous lattice-based protocols like STAG or DRAG, our protocol enhances communication rounds and guarantees scalability, making it appropriate for IoT networks. Table 2.13 Summarizes some of the parameters related to the performance and security of GKE schemes based on their security parameters. The table presents the key characteristics of each scheme, including the underlying mathematical assumption, complex problem, the security model employed, the authentication

method, and whether the scheme is analyzed as applicable for IoT networks where resource constraints and scalability are critical.

2.5.1 Security Requirements for Post-Quantum GKM Schemes:

As defined in Section 1.1.3.1, the security requirement is contingent on eight parameters (Table 2.15). In addition to these parameters, the following security features (Table 2.14) should be considered if a protocol claims to be quantum resistant.

- **Mathematical Assumption:** Quantum-resistant cryptography relies on mathematical assumptions that are considered resilient against attacks from classical and quantum computers. These assumptions are the basis for developing cryptographic protocols that maintain security even when faced with powerful quantum adversaries. (As shown in Figure 2.9)
- **Security Model:** A quantum-resistant cryptography security model provides the assumptions, features, and criteria for cryptographic protocols or schemes to secure against quantum adversaries. Security models objectively assess cryptographic constructions' weaknesses and strengths. They allow researchers to test these constructs' robustness against various attacks, including quantum computing threats.
- **Quantum Resistance Type:** The presence of quantum-resistant features in a future quantum (FQ) or post-quantum (PQ) scenario is specified [87]. In post-quantum scenarios, the adversary is assumed to possess quantum computation capabilities during the protocol's execution. On the other hand, in future quantum scenarios, the adversary can only access quantum computation after the protocol execution has concluded.

Table 2. 14 Comparison of GKM Schemes based on performance and security parameters

	GKM Schemes	Mathematical Assumption	Hard Problem	Security Model	Authentication Method	No. of Rounds	Type of Quantum-Resistance	Analysis of IoTs network
1.	STAG [84]	Lattice	Ring-LWE	ROM	x	3	PQ	x
2.	DRAG [83]	Lattice	Ring-LWE	ROM	PQ- sign	3	PQ	x
3.	n-UM [80]	Isogeny	CSIDH	QROM	CK model	1	PQ	x
4.	BC n-DH [80]	Isogeny	CSIDH	ROM	CK model	1	PQ	x
5.	CRGKE [82]	Lattice	Ring-LWE	x	x	2	PQ	x
6.	2RGKE [82]	Isogeny	CSIDH	x	x	2	PQ	x
7.	A-SIT [79]	Isogeny	SSDDH	MSU	Compiler	3	PQ	x
8.	GSIDH [81]	Isogeny	GSSCDH	x	x	n-1 (n= no. Of users)	PQ	x
9.	SIBD [81]	Isogeny	SSDDH	x	x	2	PQ	x
10.	GKEQF [88]	Compiler	DDH	x	Password	3	FQ	x
11.	PQGKE [73]	Lattice	Module-LWE	QROM	Pairwise Distributed Passwords	4	PQ	x
12.	This work	Lattice	Module-LWR	QROM	Pairwise Distributed Passwords	4	PQ	✓

Table 2. 15 Comparing Post-quantum GKM schemes in terms of security features.

No.	GKM Schemes	Forward/ Backward Secrecy	Anti-Collision	Message Confidentiality	Member Authentication	Message Integrity	Group Independence	Instant Rekey	Resistance to Quantum Attack
1.	STAG [84]	✓/✓	X	✓	X	X	✓	✓	✓
2.	DRAG [83]	✓/✓	✓	✓	✓	✓	✓	✓	✓
3.	n-UM [80]	✓/ X	X	✓	✓	✓	✓	✓	✓
4.	BC n-DH [80]	✓/ X	X	✓	✓	✓	✓	✓	✓
5.	CRGKE [82]	✓/ X	✓	✓	✓	✓	✓	✓	✓
6.	2RGKE [82]	✓/ X	✓	✓	✓	✓	✓	✓	✓
7.	A-SIT [79]	✓/X	X	✓	✓	✓	✓	✓	✓
8.	GSIDH [81]	✓/X	X	✓	X	✓	✓	X	✓
9.	SIBD [81]	✓/X	X	✓	X	✓	✓	X	✓
10.	GKEQF [88]	✓/ X	X	✓	✓	✓	✓	✓	✓
11.	PQGKE [89]	✓/✓	✓	✓	✓	✓	✓	✓	✓

2.5.2 Security Analysis for Post-Quantum GKM Schemes:

Post-quantum GKM schemes aim to strengthen communication security against the increasing threat of quantum computing. In this section, we discuss a variety of GKM landscape-fortification strategies. These schemes attempt to achieve post-quantum security while preserving the privacy and integrity of group communications. A thorough examination of these schemes reveals slight variations in their mathematical foundations, security models, and security features.

Lattice-based approaches include various schemes such as STAG, DRAG, CRGKE, and PQGKE. These schemes rely on the computational hardness of lattice problems, specifically the Ring-LWE assumption. STAG and DRAG are protocols that enable constant-round group key exchange based on the Ring-LWE assumption. The STAG protocol, implemented in the Random Oracle Model (ROM), provides resistance against passive eavesdropping attacks. However, the DRAG protocol, which is an authenticated constant-round dynamic group key exchange, modifies the key computation phase of the STAG protocol. Both schemes provide perfect forward secrecy, perfect backward secrecy, and group independence while also guaranteeing message confidentiality, member authentication, and message integrity.

The CRGKE scheme is a constant-round GKM scheme incorporating secure key derivation functions (KDFs) into the lattice-based framework. It is a cryptographic scheme based on the Ring-LWE assumption. It aims to achieve post-quantum security while also prioritizing efficient communication. This distinctive characteristic enhances the scheme's functionality and practicality, aligning it closely with the overarching objectives of lattice-based approaches.

PQGKE is a lattice-based method that utilizes the NIST-winning KEM Kyber to attain a wide range of security capabilities. It is the practical implementation of the work proposed in[87]. The system's use of Kyber, which operates within lattice-based parameters, strengthens its security against post-quantum threats. Kyber's victory in the NIST competition significantly strengthens the credibility of the lattice-based approach. PQGKE's security has been demonstrated in the Quantum Random Oracle Model (QROM), enhancing its post-quantum security credentials.

Isogeny-based GKM schemes, such as n-UM, BC n-DH, and A-SIT, utilize cryptographic invariant maps and isogeny assumptions to strengthen post-quantum security. These schemes differ in terms of their security assumptions, such as n-way decisional versus n-way gap Diffie-Hellman, and their security models, such as quantum random oracle versus random oracle. n-UM, validated in the QROM, and BC n-DH, validated in the ROM, respectively. This resilience guarantees several important security properties, including message confidentiality, member authentication, message integrity, group independence, and instant rekeying in response to member changes. A-SIT follows the Manulis-Suzuki-Ustaoglu (MSU) security model, emphasising instant rekeying and dynamic group membership administration while maintaining communication integrity.

GKEQF is a two-round password-based GAKE protocol that combines a KEM and a Message Authentication Code (MAC). This construction derives its strength from the future-quantum scenario. Security considerations only consider adversaries with quantum computation capabilities after the protocol execution is completed. GKEQF distinguishes itself by prioritizing forward secrecy and instant rekeying in response to quantum threats that extend beyond the immediate protocol interaction.

However, various quantum-resistant GKM systems show collective efforts to overcome quantum vulnerabilities. Kyber's NIST victory shows that lattice-based constructions are promising. These schemes allow enterprises to configure their security with dynamic group membership, cryptographic strength, and communication assurance. The technique chosen relies on safety requirements, the operational environment, and expected quantum challenges.

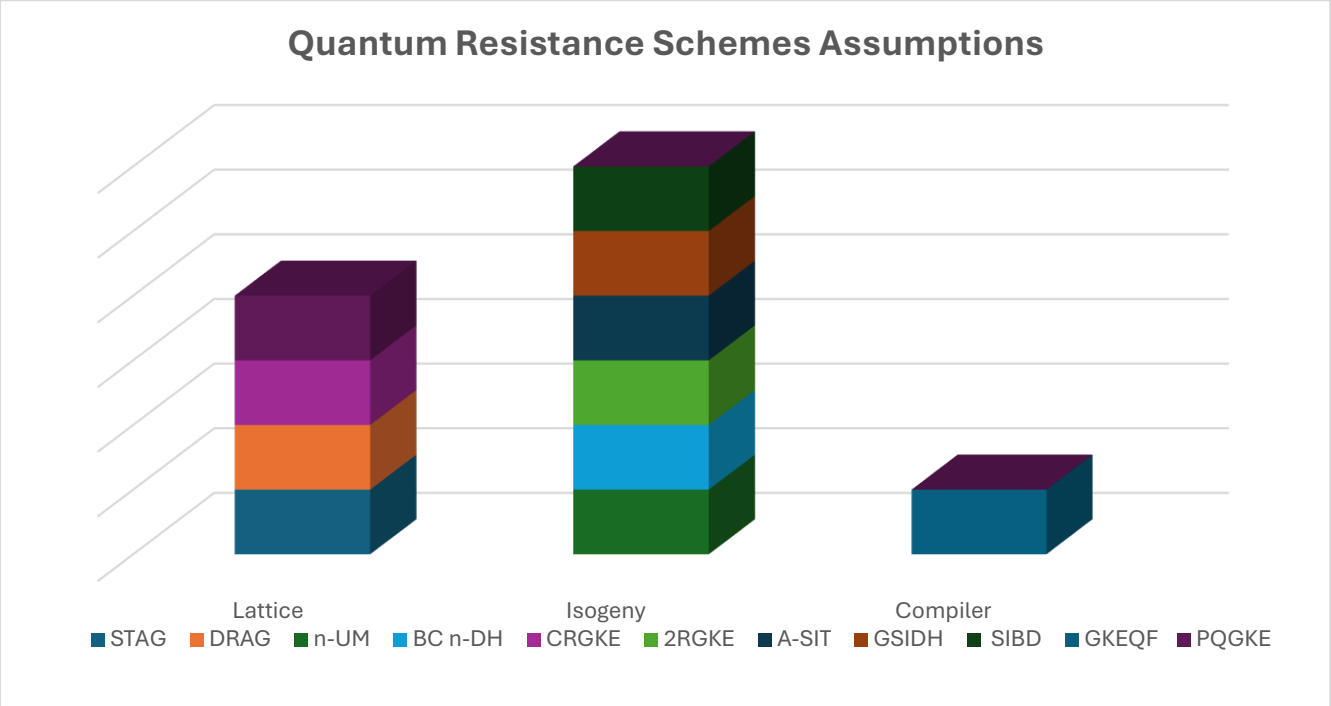

Figure 2. 6 Categories Quantum Resistance GKM schemes based on Mathematical assumptions.

2.6 Summary

This chapter addresses significant issues and research gaps in GKM for secure communication in IoT networks, focusing on the rising dangers of quantum computing. It emphasizes that most contemporary GKM techniques rely on classical cryptographic methods like RSA and ECC, which are susceptible to quantum attacks because they rely on integer factorization and discrete logarithm difficulties. The chapter emphasizes the importance of post-quantum cryptography methods that may withstand quantum

attacks while being efficient enough for resource-constrained IoT contexts. The main concern raised is the inefficiency of rekeying systems, which cause significant computational and communication overhead in dynamic networks, considering them unsuitable for large-scale IoT deployments.

The chapter also discusses scalability issues, noting that current GKM models frequently struggle to combine security and performance, particularly when handling large groups of devices. Many present methods lack enough real-world testing, emphasizing theoretical models rather than genuine IoT applications. Furthermore, the chapter highlights the importance of security aspects such as forward and backward secrecy, instant rekeying, and message integrity in quantum-resistant schemes. It emphasizes the significance of creating lightweight, scalable, and secure post-quantum GKM frameworks that can provide long-term protection against classical and quantum threats.

To establish a clear alignment between these identified research gaps and the study's direction, **Table 2.16** below maps each gap to a corresponding research objective outlined earlier in Section 1.3.2. This mapping provides a transparent bridge between the literature review and the specific goals pursued in this research.

Table 2. 16 Mapping of Identified Research Gaps to Research Objectives

Research gaps	**Research Objectives (section 1.3.2)**
Limited Literature on SGC Protocols	RQ1
Challenges in Efficient Group Key Management	RQ1
Insufficient Existing Schemes	RQ2
Lack of Quantum Resistance Considerations	RQ2

CHAPTER 3

MATHEMATICAL BACKGROUND

This chapter presents the theoretical foundation for the research, focused on cryptographic principles and transformations relevant to post-quantum security. It begins with definitions of lattice-based problems, including LWE, LWR, and Mod-LWR, and afterward delves into a comprehensive examination of the QROM. The chapter explores transformations, including techniques for obtaining IND-CCA KEMs from IND-CPA PKEs and further enhancing KEMs to IND-CCA PKEs. Furthermore, it includes Abdalla's Compiler for transitioning from two-party to group communication and the utilized cryptographic tools. Finally, the security assumptions and proof models underlying the proposed scheme are examined.

3.1 Definitions and Lattice-Based Cryptography

Lattices are geometric objects defined as intersection points of an infinite and regular n-dimensional grid. The mathematical expression of the lattices is recalled in Definition 3.1

Definition 3.1 (Lattices $\mathcal{L}$): A Lattice $\mathcal{L}$ in $\mathbb{R}^n$ (where $\mathbb{R}^n$ n-dimensional Euclidean space). Is the set of all linear combinations of a set of linearly independent basis vectors $b_1, b_2, b_3 \dots \dots b_m$ with integer coefficients in $\mathbb{Z}$:

$$\mathcal{L} = \left\{ \sum_{i=1}^{m} \mathcal{Z}_i b_i : \mathcal{Z}_i \in \mathbb{Z},\ b_i \in \mathbb{R} \right\}$$

Where: $\boldsymbol{b_1, b_2, b_3} \dots \dots \boldsymbol{b_m}$ are basis vectors spanning the $\mathcal{L}$, $\mathbb{Z}$ represent the set of integers & m is rank of $\mathcal{L}$, $m \leq n$

3.1.1 LWE, LWR and Mod-LWR problems:

Although seemingly simple lattices contain a complex combinatorial structure, they find applications in various fields, including mathematics, computer science, combinatorial optimization, and cryptography.

The mathematical foundation for post-quantum security is established by Learning with Errors (LWE) and Learning with Rounding (LWR), guaranteeing resilience against quantum adversaries. The LWE problem, proposed by Regev [90], requires differentiating between randomly chosen samples $(\boldsymbol{a}, u) \leftarrow \mathcal{U}(\mathbb{Z}_q^{l\times 1} \times \mathbb{Z}_q)$ from LWE-samples of the form,

$$(\boldsymbol{a}, b = \boldsymbol{a}^T s + e) \in \mathbb{Z}_q^{l\times 1} \times \mathbb{Z}_q$$

Where $\boldsymbol{s} \leftarrow \beta_\mu(\mathbb{Z}_q^{l\times 1})$ is a predetermined secret vector, $\boldsymbol{a} \leftarrow \mathcal{U}(\mathbb{Z}_q^{l\times 1})$ is a uniformly random vector, and $\boldsymbol{e} \leftarrow \beta_\mu(\mathbb{Z}_q)$ is a minor error. Langlois and Stehle [91] analyzed Mod-LWE, a variant of LWE that replaces the ring $\mathbb{Z}_q$ in the samples with a quotient ring $\mathcal{R}_q$. The error distribution $\beta_\mu(\mathcal{R}_q^{l\times 1})$ corresponds to the rank of the module being l, and the dimension of the ring $\mathcal{R}_q$ is $\boldsymbol{n}$. The situation when $l = 1$ corresponds to the ring-LWE problem initially presented in reference [92].

The LWR problem, as defined by Banerjee et al. [93], is a reduction variant of the LWE problem. The noise is produced in a deterministic manner by scaling and rounding coefficients from modulo q to modulo p. It is essential to note that q is more fantastic than p in this context:

$$\left(\boldsymbol{a}, b = \left\lfloor \frac{p}{q}(\boldsymbol{a}^T \boldsymbol{s}) \right\rceil\right) \in \mathbb{Z}_q^{l\times 1} \times \mathbb{Z}_q$$

The hardness assumption of the module version of LWR (Module-LWR) is a straightforward generalization of Module-LWE introduced in [94].

$$\left(\boldsymbol{a}, b = \left\lfloor \frac{p}{q}(\boldsymbol{a}^T \boldsymbol{s}) \right\rceil\right) \in \mathcal{R}_q^{l\times 1} \times \mathcal{R}_q$$

Definition 3.2 (Module-LWR assumption [94] **):** The Module-LWR Where $\boldsymbol{s} \leftarrow \beta_\mu(\mathcal{R}_q^{l\times 1})$ is a predetermined secret vector, $\boldsymbol{a} \leftarrow \mathcal{U}(\mathcal{R}_q^{l\times 1})$ is a uniformly random vector. The advantage of adversary $\mathcal{A}$ identifying m samples from a Mod-LWR distribution from that of a uniform distribution is defined as follows: $m, k, q, and\ p$ are positive integers, where $q > p$.

$$Adv_{m,l,\mu,q,p}^{Mod-LWR}(A) = \left| \begin{matrix} \Pr\left(b' = 1: \begin{matrix} \boldsymbol{A} \leftarrow \mathcal{U}(\mathcal{R}_q^{m\times l}); \boldsymbol{s} \leftarrow \beta_\mu(\mathcal{R}_q^{l\times 1}); \\ b' = A(\boldsymbol{A}, \lfloor (p/q)\boldsymbol{A}\boldsymbol{s} \rceil); \end{matrix}\right) \\ -\Pr\left(b' = 1; \begin{matrix} \boldsymbol{A} \leftarrow \mathcal{U}(\mathcal{R}_q^{m\times l}); \boldsymbol{u} \leftarrow \beta_\mu(\mathcal{R}_p^{l\times 1}); \\ b' = A(A, \boldsymbol{u}); \end{matrix}\right) \end{matrix} \right|$$

3.2 Quantum Computation

Quantum computation exploits the properties of quantum mechanics, such as superposition, entanglement, and interference, to perform computations that are infeasible for classical systems. This section presents essential principles, fundamental definitions, and significant theorems pertinent to quantum computation and its ramifications in cryptography, especially within the Quantum Random Oracle Model (QROM).

Definition 3.3 (Qbit and Superposition): A qubit $|\varphi\rangle$, the fundamental unit of quantum information, is a quantum state in a two-dimensional complex vector space $\mathbb{C}^2$. It can be represented as:

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$$

Where, $\alpha, \beta \in \mathbb{C}$ satisfy $|\alpha|^2 + |\beta|^2 = 1$

Here, $|\alpha|^2$ and $|\beta|^2$ are the probabilities of measuring the qubit in states $|0\rangle$ and $|1\rangle$, respectively.

Definition 3.4 (Quantum Register): A quantum register is a system of n-qubit,

Hibert space: represented as a state $|\varphi\rangle = \sum_{x\in\mathbb{F}_2^n} \alpha_x |x\rangle$,

where α_x are a complex amplitudes such that $\sum_{x\in\mathbb{F}_2^n}|\alpha_x|^2 = 1$.

This is the superposition that allows quantum computers to process many states simultaneously.

Definition 3.5 (Measurement): Measurement in quantum mechanics induces the collapse of a qubit or quantum register's superposition into one of its basic states. For a qubit $|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$, measurement in the standard basis results in outcome 0 with a probability of $|\alpha|^2$ and outcome 1 with a probability of $|\beta|^2$.

Definition 3.6 (Quantum Oracle): A quantum oracle is a mechanism employed in quantum algorithms to assess a function on superpositions of inputs. It functions as follows:

$$|x\rangle|y\rangle \mapsto |x\rangle|y \oplus O(x)\rangle$$

where $O : \mathbb{F}_2^n \to \mathbb{F}_2^m$ is the function under evaluation.

3.2.1 Quantum Random Oracle Model (QROM):

The QROM extends the classical Random Oracle Model (ROM) by allowing quantum adversaries to query oracles in superposition. While classical adversaries query a random oracle O on individual inputs x, a quantum adversary can submit a quantum state:

Definition 3.7: Let $O : \{0,1\}^n \rightarrow \{0,1\}^m$ be a random oracle. A quantum adversary can query O in superposition:

$$| \varphi\rangle = \sum_{x\epsilon\{0,1\}^n} \alpha_x | x\rangle, \qquad where \sum |\alpha_x|^2 = 1$$

The oracle response:

$$O(| \varphi\rangle) = \sum_{x\epsilon\{0,1\}^n} \alpha_x | x, O(x)\rangle,$$

Where $O(x)$ is the randomly chosen output of the input x. This enhanced capability requires cryptographic schemes to account for superposition queries, quantum search algorithms, and oracle consistency across states.

Theorem 3.1(Zhandry result): No quantum algorithm making at most q queries to a quantum oracle can differentiate between a genuinely random function. $O : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$ and a 2q-wise independent function f_{2q}.

Proof of Theorem:

We demonstrate this theorem by utilising the characteristics of 2q-wise independent functions, which guarantee that any 2q inputs produce outputs indistinguishable from those of a genuinely random function.

Step 1: Configuration

Examine a quantum adversary A that submits q enquiries to a quantum oracle $|\ O\ \rangle$. Every inquiry exists in a superposition of classical inputs.

Step 2: Definition of 2q-wise Independent Function

A function f_{2q} is 2q-wise independent if, for every subset of up to 2q inputs, the outputs are distributed similarly to those of a random function.

Step 3: Discriminative Experiment

Allow the oracle to execute either a random function O or a 2q -wise independent function f_{2q}. The adversary's objective is to differentiate between these two scenarios based on the observed outputs.

Step 4: Bounding the Distinguishing Probability

The probability that the opponent can differentiate between O and f_{2q} is constrained by the statistical distance between the output distributions generated by O and f_{2q}. This distance is insignificant owing to the 2q-wise independence of f_{2q}.

The Advantage of A in differentiating the two scenarios is expressed by:

$$\left|Pr[1 \leftarrow A^{|O\rangle}] - Pr[1 \leftarrow A^{|f2q\rangle}]\right| \leq \epsilon$$

where ϵ is a negligible quantity contingent upon q and the function's parameters.

Step 5: Conclusion

Given that A's capability to differentiate O from f_{2q} is minimal, the theorem is established, illustrating that q enquiries are inadequate for any quantum algorithm to differentiate between a random function and a 2q-wise independent function.

3.3 Quantum Secure Authenticated Key Exchange (AKE):

An Authenticated Key Exchange (AKE) is a cryptographic protocol designed to enable multiple participants to establish a shared secret key while simultaneously ensuring the authenticity of each involved party. This mechanism safeguards against unauthorized entities impersonating legitimate participants, thus maintaining the confidentiality and integrity of the key. A Quantum Secure AKE enhances this concept to protect against adversaries utilizing quantum computing capabilities, which present a risk to numerous classical cryptographic systems. Utilizing post-quantum security principles, quantum-secure AKE's guarantee the security of the shared session key, even against quantum-capable attackers.

Hövelmanns et al, proposed a FO_{AKE} (Fujisaki-Okamoto for AKE), a general framework for two-message AKE protocols. This construction transforms any passive secure public-key encryption (PKE) scheme into an actively secure AKE. Unlike previous methods, FO_{AKE} doesn't require the PKE scheme to have perfect correctness. This flexibility enables the incorporation of lattice- or code-based cryptography methods with small correctness errors. Furthermore, FO_{AKE} eliminates the requirement for digital signature schemes, which tend to be computationally intensive. FO_{AKE} is especially advantageous for post-quantum cryptographic protocols.

3.3.1 Transformation from IND-CPA PKE to Secure 2-AKE

In this subsection, we focus on transforming a secure PKE method under IND-CPA into a secure 2-AKE protocol. The transformation tackles the issue of attaining active security within the QROM based on a passively safe encryption approach. The structure employs techniques like key puncturing and hash functions to safeguard against active assaults, hence assuring the protocol attains IND-StAA. This security architecture

guarantees that the session key remains indistinguishable from random, even if an intruder alters messages or compromises partial secret information.

The transformation typically entails altering the foundational PKE scheme to accommodate decryption failures and imperfect correctness, as numerous lattice-based and code-based encryption schemes display slight accuracy deficiencies. Through the rigorous arrangement of the encryption and decryption processes and the incorporation of further randomness via hash-based transformations, the resultant 2-AKE protocol can be secure against both classical and quantum adversaries.

3.3.1.1 Public-key Encryption:

Public-key encryption (PKE) acts as a fundamental cryptographic method that facilitates secure communication over untrusted channels by differentiating encryption and decryption processes using two separate keys. The process consists of three main algorithms: Key Generation (KG), Encryption (Enc), and Decryption (Dec). These algorithms function collaboratively to maintain data confidentiality. The key generation algorithm generates a pair of cryptographic keys: a public key (pk) utilised for encryption and a corresponding secret key (sk) employed for decryption. The public key defines the randomness space R(pk) and the ciphertext space C. Encryption utilises the public key to convert a message m m from a specified message space M M into a ciphertext c c, effectively obscuring the contents of the message. The ciphertext is contingent upon values that are randomly selected from the randomness space R. When the randomness utilised in the encryption process is clearly defined, the encryption can be represented as $c := Enc(pk, m; r)$. The decryption algorithm necessitates the use of the secret key to reverse the encryption process and recover the original message. Upon verification of the ciphertext's validity and proper structure, the decryption function will

yield the original message. If the ciphertext is deemed invalid or has been altered, the decryption process will yield a specific error symbol ⊥, signifying a decryption failure.

PKE systems require assessment of their functional correctness and security guarantees to ensure reliability. Functional correctness guarantees that when a message undergoes encryption and subsequent decryption using the appropriate keys, there is a high probability that the original message will be accurately retrieved. Security guarantees concentrate on safeguarding encrypted data from unauthorised decryption, regardless of the computational resources available to potential adversaries. A variety of formal definitions are utilised to evaluate the strength and reliability of public-key encryption systems.

Definition 3.8 (Collision probability of key generation): It quantifies the probability that two independently generated key pairs will yield identical public keys. This metric is critical as it signifies the potential risk associated with multiple users utilising the same public key, which may lead to security vulnerabilities. We define

$$\mu(KG) = Pr[(pk, sk) \leftarrow KG, (pk', sk') \leftarrow KG : pk = pk']$$

A low collision probability is essential for the robustness of key generation process.

Definition 3.9 (Ciphertext collision probability): It is a measure of the probability that two distinct messages will generate identical ciphertexts when encrypted under the same public key. We define as,

$$\mu(Enc) := Pr[(pk, sk) \leftarrow KG, m, m' \xleftarrow{\$} M, c \leftarrow Enc(pk, m), c' \leftarrow Enc(pk, m') : c = c']$$

This property is crucial because it influences the uniqueness of ciphertexts and, as a result, the scheme's capacity to withstand specific cryptographic attacks. A secure scheme is intended to reduce the probability of ciphertext collisions, guaranteeing that individual messages generate distinct ciphertexts.

Definition 3.10 (Spreadness property): It referred to as γ-spreadness, which guarantees that ciphertexts demonstrate a uniform distribution, rendering it difficult to predict. A PKE scheme is defined as γ-spread when the maximum likelihood of a particular ciphertext arising from a specified message and public key is constrained by $2^{-\gamma}$. We state that PKE is γ-spread iff for all key pairs (pk,sk) ∈ supp(KG) and all messages m ∈ M it holds that

$$\max_{c \in C} Pr[r \xleftarrow{\$} \mathcal{R}: Enc(pk, m; r) = c] \leq 2^{-\gamma}$$

This concept plays a vital role in probabilistic encryption schemes, wherein randomness is employed throughout the encryption process. This characteristic mitigates the probability of ciphertext predictability, even when some details regarding the technique used for encryption are accessible.

Definition 3.11 (Correctness): Correctness is quantified by a decryption error probability δ, where δ is represents the probability that a valid ciphertext fails to decrypt correctly:

$$\delta = E[max_{m \in M} Pr[c \leftarrow Enc(pk, m): Dec(sk, c) \neq m]]$$

In certain encryption schemes, particularly post-quantum cryptography systems, there exists a minor probability of decoding failure attributable to deficiencies in the

encryption algorithm's mathematical framework. A PKE technique is considered reliable if the decryption error probability δ is insignificantly small.

Next, we reproduce the results given in [95] about the IND-StAA security of the $\mathrm{FO}_{\mathrm{AKE}}$ transformation. The theorem presented asserts that the IND-StAA security of $\mathrm{AKE} = \mathrm{FO}_{\mathrm{AKE}}(\mathrm{PKE}, \mathrm{G}, \mathrm{H})$ is contingent upon the DS and IND-CPA security of the PKE scheme, where PKE denotes a public key encryption scheme and G and H represent random oracles.

GAME $IND - CPA_b$

01 $(pk, sk) \leftarrow KG$
02 $(m*0, m*1, st) \leftarrow A_1(pk)$
03 $c^* \leftarrow Enc(pk, m_b^*)$
04 $b' \leftarrow A2(pk, c^*, st)$
05 $return\ b'$

Definition 3.12 (IND-CPA secure PKE [95]**):** Let $\mathrm{PKE} = (\mathrm{KG}, \mathrm{Enc}, \mathrm{Dec})$ to be a PKE scheme. The security of the scheme is measured based on the (IND-CPA) notion as define in **GAME $IND - CPA_b$**. The advantage of a quantum adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ in relation to the IND-CPA, the security of PKE is defined as follows:

$$Adv_{PKE}^{IND-CPA}(\mathcal{A}) := |Pr[IND - CPA_1^{\mathcal{A}} \Rightarrow 1] - Pr[IND - CPA_0^{\mathcal{A}} \Rightarrow 1]|$$

Definition 3.13 (DS [95]**):** Let $PKE = (KG, Enc, Dec)$ be a PKE scheme with message space $\mathcal{M}$ and ciphertext space C, coming with an additional PPT algorithm $\overline{Enc}$. For quantum adversaries $\mathcal{A}$, we define the advantage against PKE's disjoint simulatability as

$$Adv^{DS}_{PKE,\overline{Enc}}(\mathcal{A}) \coloneqq \left| \Pr\left[\begin{matrix} pk \leftarrow KG \\ m \overset{\$}{\leftarrow} \mathcal{M} \\ c \leftarrow Enc(pk,m) \end{matrix} : 1 \leftarrow \mathcal{A}(pk,c) \right] - \Pr\left[\begin{matrix} pk \leftarrow KG \\ c \leftarrow \overline{Enc}(pk) \end{matrix} : 1 \leftarrow \mathcal{A}(pk,c) \right] \right|$$

When there is no chance of confusion, we will drop Enc from the advantage's subscript for convenience. We call PKE ϵdis-disjoint if for all $pk \in$ supp (KG),

$$Pr[c \leftarrow \overline{Enc} : c \in Enc(pk, M; R)] \leq \epsilon dis$$

where $R = R(pk)$ is a finite randomness space defined by pk.

The Construction. In a public-key encryption scheme $PKE_0 = (KG_0, Enc_0, Dec_0)$ with a message space $\mathcal{M}_0$, we can associate $PKE \coloneqq Punc[PKE_0, \hat{m}] \coloneqq (KG \coloneqq KG_0, Enc, Dec \coloneqq Dec_0)$ with a message space $\mathcal{M} \coloneqq \mathcal{M}_0 \backslash \{\hat{m}\}$, where $\hat{m}$ is a message in $\mathcal{M}$. The definitions of encryption and fake encryption sampling of PKE can be found in

$\underline{\boldsymbol{Enc(pk, m \in M)}}$	$\underline{\boldsymbol{\overline{Enc}(pk)}}$
$\boldsymbol{01\ c \leftarrow Enc_0(pk, m)}$	$\boldsymbol{01\ c \leftarrow Enc_0(pk, \hat{m})}$
$\boldsymbol{02\ return\ c}$	$\boldsymbol{02\ return\ c}$

Theorem 3.3([95]): Let PKE = (KG, Enc, Dec) represent a PKE scheme that exhibits $(1 - \delta)$ correctness. Furthermore, the scheme is assumed to include a sampling algorithm $\overline{Enc}$ that is ε-disjoint. Let N represent the number of parties. It is assumed that an attacker can access a REVEAL oracle, which discloses the corresponding session key if it has already been established. For any IND-StAA adversary $\mathcal{B}$ that establishes S sessions, and issues a maximum of q_R (classical) queries to REVEAL, a

maximum of q_G (quantum) queries to random oracle G, and a maximum of q_H (quantum) queries to random oracle H, there exist, adversaries, $\mathcal{A}_{DS}$ and $\mathcal{A}_{CPA}$ against the PKE scheme such that:

$$Adv_{AKE}^{IND-StAA}(\mathcal{B}) \leq 2S(S+3N)Adv_{PKE}^{DS}(\mathcal{A}_{DS}) + 4S(S+3N)\sqrt{(q_G+2q_H+3S)Adv_{PKE}^{cpa}(A_{CPA}) + {4(q_G+2q_H+3S)^2}/{|M|}} + 32(S+3N)(q_G+2q_H+3S)^2(1-\delta) + 4S(S+N)\,\varepsilon_{dis} + S^2(N+1)\,\mu(KG)\mu\,(Enc) + 2S^2 + \mu(KG)$$

and the running times of $\mathcal{A}_{DS}$ and $\mathcal{A}_{CPA}$ is about that of $\mathcal{B}$. Here,

$$\mu(KG) = \Pr\left[(pk, sk) \leftarrow KG\,, (pk', sk') \leftarrow KG : pk = pk'\right]$$

And

$$\mu(Enc) = \Pr\left[(pk, sk) \leftarrow KG\,, m, m' \leftarrow \mathcal{M}, c \leftarrow Enc(pk, m), c' \leftarrow Enc(pk, m') : c = c'\right]$$

3.3.2 Transformation from IND-CPA PKE to IND-CCA KEM

In this subsection, we investigate the transformation process for converting an IND-CCA-secure KEM into an IND-CCA-secure PKE scheme. This transformation serves as a fundamental element in contemporary cryptographic architecture, facilitating secure message encryption and utilising the modular characteristics of KEMs. The integration of the encapsulated key with a symmetric encryption mechanism results in a PKE scheme that retains the robustness of the underlying KEM, thereby providing security against adaptive chosen ciphertext attacks.

3.3.2.1 Key Encapsulation Mechanism (KEM):

A Key Encapsulation Mechanism (KEM) in cryptography comprises three fundamental algorithms: key generation, encapsulation, and decapsulation. The key generation algorithm (KG) produces a key pair that includes a public key (pk) and a secret key (sk), with the public key also defining a finite key space (κ). The encapsulation algorithm (Encaps) accepts a public key as input. It generates a tuple (K, c), where K represents a randomly generated key, and c denotes its encapsulation within the key space. The decapsulation algorithm (Decaps) is deterministic; it takes a secret key and an encapsulation c as inputs and outputs either the encapsulated key K or a special symbol $\perp \notin \kappa$ to indicate that c is not an invalid encapsulation.

Correctness is an essential component of KEMs. A KEM is classified as δ-correct if the probability of decapsulation using the secret key and encapsulation not producing the original key does not exceed δ. This property guarantees the reliability of the mechanism, especially in the random oracle model, where ciphertexts are independent of messages.

$$Pr[Decaps(sk, c) = K \mid (pk, sk) \leftarrow KG; (K, c) \leftarrow Encaps(pk)] \leq \delta$$

The aim of KEMs is to ensure IND-CCA. This security concept guarantees that an adversary, even with access to a decapsulation oracle and the ability to make adaptive queries, is unable to differentiate between a genuine key encapsulation and a random one. The IND-CCA game assesses the adversary's ability to distinguish between scenarios, with the advantage quantified as the absolute difference between the adversary's success probability and 0.5, indicative of a random guess.

GAME IND-CCA

$01.(pk, sk) \leftarrow KG$
$02.b \leftarrow_{\$} F2$
$03.(K_0^*, c^*) \leftarrow Encaps(pk)$
$04.K_1^* \leftarrow_{\$} K$
$05.b' \leftarrow \mathcal{A}^{Decaps}(pk, c^*, K_b^*)$
$06.return\ b' = b$

$Decaps(c \neq c^*)$

$07.K := Decaps(sk, c)$
$08.return\ K$

Definition 3.14 (KEM IND-CCA security): In the IND-CCA game, the adversary is challenged to distinguished between the encapsulation of a real key and random key. The game output a success if the adversary $\mathcal{A}$ correctly identifies whether the key encapsulation corresponds to the real or random key, and the IND-CCA advantage for an adversary $\mathcal{A}$ against a KEM is state as:

$$Adv_{KEM}^{IND-CCA}(\mathcal{A}) := |Pr[IND - CCA^{\mathcal{A}} \Rightarrow 1] - 1/2|.$$

We reproduce the results given in [95] about IND-CCA security of the $\mathrm{FO}_m^{\not\perp}$transformation, as presented in [95], is reproduced herein. The theorem presented asserts that the IND-CCA security of $\mathrm{FO}_{\mathrm{m}}^{\not\perp} = \mathrm{FO}^{\not\perp}(\mathrm{PKE}, \mathrm{G}, \mathrm{H})$ is contingent upon the DS and IND-CPA security of the PKE scheme, where PKE denotes a public key encryption scheme and G and H represent random oracles.

Theorem 3.4([95]): Let PKE = (KG, Enc, Dec) represent a PKE scheme that exhibits $(1 - \delta)$ correctness. Furthermore, it is assumed that the scheme includes a sampling algorithm $\overline{Enc}$ that is ε-disjoint. It is assumed that an attacker has access to a DECAPS oracle, then any (quantum) IND-CCA adversary $\mathcal{A}$, issues a maximum of q_D (classical) queries to DECAPS oracle, a maximum of q_G (quantum) queries to a random oracle G,

and a maximum of q_H (quantum) queries to a random oracle H, there exist adversaries, $\mathcal{B}_{DS}$ and $\mathcal{B}_{CCA}$ against the PKE scheme such that:

$$Adv_{KEM}^{IND-CCA}(\mathcal{A}) \leq 8\,(2q_G + q_H + q_D + 4)^2\delta + Adv_{PKE}^{DS}(\mathcal{B}_{DS}) + 2\sqrt{(q_G + q_H)Adv_{PKE}^{IND-CPA}(\mathcal{B}_{IND-CCA}) + {4(q_G + q_H)^2}/{|M|}} + \varepsilon_{dis}$$

and the running times of $\mathcal{B}_{DS}$ and $\mathcal{B}_{IND-CPA}$ is about that of $\mathcal{A}$.

3.3.3 Transformation from IND-CCA KEM to IND-CCA PKE

The transformation of an IND-CCA-secure KEM into an IND-CCA-secure PKE scheme represents a significant development in cryptography. This approach connects key encapsulation with complete message encryption through the modularity inherent in cryptographic primitives. This transformation aims to facilitate secure communication in situations necessitating public key encryption.

Definition 3.15 (PKE IND-CCA security): A PKE is IND-CCA secure if no polynomial-time adversary $\mathcal{A}$, with access to a decryption oracle, can distinguish the encryption of two chosen plain text m_0 and m_1 .

$$\left| Adv_{IND-CCA}^{PKE}(\mathcal{A}) = Pr[\mathcal{A}(pk, C^*) = b] - \frac{1}{2} \right|$$

Where $C^* = Enc(pk, m_b)$ $b \in \{0,1\}$ is chosen at random.

Theorem 3.5: Let $\mathrm{KEM} = (\mathrm{KG}, \mathrm{Encaps}, \mathrm{Decaps})$ be a KEM that is secure under adaptive chosen ciphertext attacks (IND-CCA) and let $\mathrm{SKE} = (\mathrm{Enc}, \mathrm{Dec})$ be a symmetric encryption scheme that is one-time secure used in DEM (Data Encapsulation

Mechanism) is IND-CCA secure, then the produced PKE scheme is also IND-CCA secure.

Proof: Let $\mathcal{A}$ be an adversary against PKE scheme, we construct a sequence of games to show that $\mathcal{A}$ cannot distinguish the encryption of two chosen plain text m_0 and m_1 .

Game 0: Real Attack

1. $(pk, sk) \leftarrow KeyGen(1^{\lambda})$

2. Adversary Input:

- Provide public key pk to adversary $\mathcal{A}$.

3. Challenge Phase:

- Adversary chooses two messages m_0 and m_1

- Random bit $b \in \{0,1\}$

- Encapsulate key: $(c_1, K) \leftarrow Encaps(pk)$.

- Encrypt message: $c_2 \leftarrow EncDEM(\mathrm{K}, mb)$.

- Ciphertext: $C^* \leftarrow (c_1, c_2)$.

4. Adversary Output:

- $\mathcal{A}$ outputs guess b'.

5. Goal:

- Adversary wins if $b' = b$.

Game 1: Replace Encapsulated Key K with random key K'

1. $(pk, sk) \leftarrow KeyGen(1^{\lambda})$

2. Adversary Input:

- Provide public key pk to adversary $\mathcal{A}$.

3. Challenge Phase:

- Adversary chooses two messages m_0 and m_1 .

- Random bit $b \in \{0,1\}$.

- Replace encapsulated key:

$(c_1, K') \leftarrow Encaps(pk)$, but K' is a random key.

- Encrypt message: $c_2 \leftarrow EncDEM(K', mb)$.

- Ciphertext: $C^* \leftarrow (c_1, c_2)$.

4. Adversary Output:

- $\mathcal{A}$ outputs guess b'.

5. Security Transition:

- By IND-CCA security of KEM, $\mathcal{A}$cannot distinguish K from K'.

- Advantage: $|Pr[b' = b \text{ in } Game\ 1] - Pr[b' = b \text{ in } Game\ 0]| \approx 0$.

Game 2: Replace DEM Ciphertext c_2 with Random

1. $(pk, sk) \leftarrow KeyGen(1^{\lambda})$

2. Adversary Input:

- Provide public key pk to adversary $\mathcal{A}$.

3. Challenge Phase:

- $\mathcal{A}$ chooses two messages m_0 and m_1 .

- Random bit $b \in \{0,1\}$.

- Replace encapsulated key:

$(c_1, k') \leftarrow Encaps(pk)$, where K' is random.

- Replace DEM encryption with random:

$c_2 \leftarrow RandomString(|c_2|)$.

- Ciphertext: $C^* \leftarrow (c_1, c_2)$.

4. Adversary Output:

- $\mathcal{A}$ outputs guess b'.

5. Security Transition:

- By IND-CPA security of DEM, $\mathcal{A}$ cannot distinguish c_2 from random.

- Advantage: $|Pr[b' = b \text{ in } Game\ 2] - Pr[b' = b \text{ in } Game\ 1]| \approx 0$.

Game 3: Replace Entire Ciphertext C^* with Random

1. $(pk, sk) \leftarrow KeyGen(1^{\lambda})$

2. Adversary Input:

- Provide public key pk to adversary $\mathcal{A}$.

3. Challenge Phase:

- Adversary chooses two messages m_0 and m_1 .

- Random bit $b \in \{0,1\}$.

- Replace ciphertext:

$C^* \leftarrow RandomString(|C^*|)$.

4. Adversary Output:

- $\mathcal{A}$ outputs guess b'.

5. Security Transition:

- Adversary cannot distinguish real ciphertext from random due to IND-CCA security of KEM and IND-CPA security of DEM.

Advantage: $Pr[b' = b \text{ in } Game\ 3] = 1/2$

So, the mentioned game sequence systematically transitions from a real-world attack on the PKE scheme to a scenario in which the ciphertext is entirely random, making it

indistinguishable from noise to the adversary. The adversary interacts with the actual encryption scheme in Game 0, which represents the genuine attack. In Game 1, the encapsulated key K is substituted with a random key K', thereby ensuring that the adversary is unable to differentiate between the original and random keys by utilising the IND-CCA security of the KEM. Game 2 expands upon this by substituting the DEM (Data Encryption Mechanism) ciphertext c_2 with a random string, thereby utilising the IND-CPA security of the DEM to argue indistinguishability. Ultimately, in Game 3, the adversary is unable to differentiate between genuine ciphertext and random noise as the entire ciphertext C^* is substituted with a random string. The transitions between games are based on well-established cryptographic assumptions, including the IND-CCA security of KEM and the IND-CPA security of DEM. Collectively, these assumptions demonstrate that the adversary's advantage is reduced to negligible levels, thereby demonstrating the overall IND-CCA security of the PKE scheme.

3.4 Two-party to Group: Abdalla's Compiler:

In this section, we describe the Abdalla et al. Compiler, a method that transforms an arbitrary two-party authenticated key exchange (2-AKE) protocol into a group authenticated key exchange (GAKE) protocol. By employing fundamental cryptographic tools, including universal hash functions, pseudorandom functions (PRFs), and non-malleable commitments, the compiler converts a 2-AKE protocol into a scalable solution for secure group communication. The compiler guarantees that the emerging protocol retains scalability and efficiency while achieving robust security properties, including forward secrecy and key integrity.

3.4.1 Purpose and Overview:

Let $\mathcal{N}$ (assumed to be polynomial size) is a set of users involved in the GAKE protocol.

$$\mathcal{G} = \{\mathcal{U}_0, \mathcal{U}_1, \mathcal{U}_2 \dots \dots . \mathcal{U}_{n-1}\} \subset \mathcal{N} \text{ and } n > 2.$$

Abdalla et al.'s compiler's primary goal is to facilitate the secure agreement of a group of participants $\mathcal{G}$ on a shared session key. The underlying 2-AKE protocol pairwise authenticated key exchange is extended to a group setting to achieve this. Notably, the compiler does not require additional authentication mechanisms beyond those built into the 2-AKE protocol. Also, if 2-AKE necessitates r rounds of communication, then GAKE necessitates $r + 2$ rounds. It is presumed that the long-term secrets necessary for 2AKE have been established during a trusted authentication phase, as 2AKE is an authenticated key establishment protocol.

In a 2-AKE protocol, it is assumed that a trusted authentication phase has already established the long-term secrets required for the protocol. Depending on the setup, this phase can follow one of three cases:

- **Public and Private key Pairs:**

Each participant $\mathcal{U}_i \in \mathcal{G}$ possess a unique key pair compromising a public key pk_i and a private key sk_i. All necessary public keys are distributed to the protocol participants during the initialization phase.

- **Symmetric key Sharing:**

Every pair of participants $\mathcal{U}_i, \mathcal{U}_j \in \mathcal{G}$ $(i \neq j)$ either shares a distinct high-entropy symmetric key or the entire group $\mathcal{G}$ relies on a single shared secret key. Different protocol instances for a participant may use separate long-term secrets.

- **Password-Based Authentication:**

Each pair of participants $\mathcal{U}_i, \mathcal{U}_j \in \mathcal{G}$ $(i \neq j)$ shares a low-entropy password. In this case, it is the case, and it is assumed that passwords are selected uniformly at random from a publicly available dictionary $\mathcal{D} \subseteq \{0,1\}^*$.

3.4.2 Cryptographic Tools:

Abdalla's compiler leverages three essential cryptography primitives to ensure security and efficiency in its operation. These cryptographic tools are essential to the compiler's design, ensuring security, non-malleability, and pseudo-randomness. Collectively, they facilitate the development of a robust post-quantum group authenticated key exchange (GAKE) protocol derived from a streamlined version of the generic compiler.

1. **Non-Interactive Non-Malleable Commitment Scheme ($\boldsymbol{C}$):** A perfectly binding commitment scheme is adopted that achieves non-malleability across multiple commitments. This guarantees that an adversary who tries to alter or forge commitments cannot influence the protocol's output.
2. **Collison-Resistant Pseudorandom Family ($\boldsymbol{\mathcal{F}} = \{\mathrm{F}_{\boldsymbol{\ell}}\}_{\boldsymbol{\ell} \in \mathbb{N}}$):** The compiler uses a pseudorandom function family indexed by a set $\{0,1\}^L$ of polynomial size. Two publicly known values v_0 and v_1, are utilized such that no probabilistic polynomial-time (PPT) adversary can find two different indices $\lambda \neq \mu$ from $\{0,1\}^L$ that satisfy:

$$\mathrm{F}_{\lambda}^{\ell}(v_j) = \mathrm{F}_{\mu}^{\ell}(v_j), \qquad for\ j \in \{0,1\}$$

This property ensures that the pseudorandom function remains collision-resistant and secure.

3. **Hash Function ($\boldsymbol{H}$):** A hash function is chosen from a family of universal hash functions. This function maps the concatenation of bitstrings $\{0,1\}^{kn}$ and the participant set $\mathcal{G}$ onto $\{0,1\}^L$,where n represent the number of participants and $k \in$

$\mathbb{N}$. The hash function ensures efficient and secure mapping of input data for cryptographic computations.

Round $1 \sim r$: : Execution: Each participant U_i (where $i\ ranges\ from\ 0\ to\ n-1$), perform $2AKE$ operation with U_i and $U_{[i+1]}$ where $[k] = k\ mod\ n$ (i.e., all indices should be taken cyclically). Thus each $U_i \in \mathcal{G} = \{U_0, U_1, \ldots, U_{n-1}\}$ maintains two pairwise keys $\vec{K}_i$ and $\overleftarrow{K}_i$ shared with $U_{[i+1]}$ and $U_{[i-1]}$. Note that, $for\ i = 0, \ldots, n-1$, $\vec{K}_i = \overleftarrow{K}_{[i+1]}$.

Round $r+1$:

- Calculation: Each U_i computes $X_i = \vec{K}_i \oplus \overleftarrow{K}_i$, and selects a random r_i to derive commitment $C_i = C(i, X_i; r_i)$,
- Broadcast: Each U_i telecasts the message:

$$M_i^1 = (U_i, C_i)$$

Round $r+2$:

- Broadcast: Each U_i telecasts the message:

$$M_i^2 = (U_i, X_i, r_i)$$

- Verification: Each U_i verifies the condition $X_0 \oplus X_1 \oplus X_2 \ldots \oplus X_{n-1} = 0$ and ensures the correctness of the commitments. If any check fails, U_i terminates protocol execution setting $acci: = FALSE$. If all checks pass, U_i $acc_i := TRUE$ and $pid_i := \mathcal{G}$.
- Computation: Each U_i computes $(n-1)$ values:
  $K_{i-j} := \overleftarrow{K}_i \oplus X_{i-1} \oplus \ldots \oplus X_{i-j}$, for $j = 1,2, \ldots, n-1$.
  Next, U_i defines a master key K as:

$$K = \left(\overleftarrow{K}_0, \overleftarrow{K}_1 \ldots \ldots, \overleftarrow{K}_{n-1}, \mathcal{G}\right)$$

and sets the session key and session identifier as

$$sk_i: = \mathrm{F}_{H(K)}^{\ell}(\upsilon_1) \text{ and } sid_i := \mathrm{F}_{H(K)}^{\ell}(\upsilon_0)$$

where $\ell \in \mathbb{N}$ is the security parameter.

Figure 3. 1 Abdalla et al Compiler

3.5 Security Assumptions and Proof Models

We have adopted the security model from Abdalla et al. [78]. However, our scenario is more specific. Unlike in [78], where all the proofs based on common reference string model, our proofs operate in (quantum) random oracle model. With utmost precision, we assumed all the necessary public keys and the parameters for applying $Saber'.2AKE$ and $Saber'.PKE$ are publicly known and certified. Additionally, the description of all the hash functions involved is also made available, idealized as random oracles. Furthermore, it is presumed that all prospective protocol participants are generated and distributed for the long-term k required for authentication in

$Saber'.2AKE$ during the trustworthy setup phase. Using variables to define user-specified data storage for each protocol execution is standard practice in computer science. On top of that, oracles are used to simulate adversarial actions.

1. **Protocol Instances** [78]**:** Every participant $U_i \in \mathcal{U}$ in the protocol can run multiple instances simultaneously. The execution of a process by a protocol participant U_i can be represented by a single instance$\prod_{i}^{s_i}$. In all cases, instance s_i of protocol participant $U_i \in \mathcal{U}$ is denoted as $\prod_{i}^{s_i}$ $(i \in \mathbb{N})$. Every instance is given seven variables:

Table 3. 1 Protocol instances

Instances	Description
$\mathbf{used_i^{s_i}}$	A protocol run may have utilised this instance, and it can tell you that. Only protocol messages received by the instance because of a call to the Execute- or Send-oracle (see to below) can set the $\mathrm{used}_i^{s_i}$ flag.
$\mathbf{state_i^{s_i}}$	It keeps the long-term authentication keys and all the relevant state information during the protocol execution.
$\mathbf{term_i^{s_i}}$	Indicates whether the execution has "terminated".
$\mathbf{sid_i^{s_i}}$	denotes that there is a public session identifier which can be used to identify the session key $\mathrm{sk}_i^{s_i}$**.** It is crucial to note that adversary can still learn all session IDs, given that we don't generate them like session transcripts.
$\mathbf{pid_i^{s_i}}$	The set of identities of the individuals that s_i intends to generate a key, as well as U_i itself, is stored.
$\mathbf{acc_i^{s_i}}$	Determines success of protocol instance, i.e, user agreed on session key.
$\mathbf{sk_i^{s_i}}$	Once session key is accepted by s_i, it is stored. Prior to being accepted, it stores a distinct NULL value.

2. **Communication Network** [78]**:** We can assume that users have arbitrary connections that allow them to communicate directly. The network operates non-privately and fully asynchronously, allowing the adversary to freely eavesdrop, delay, delete, and insert messages.
3. **Adversary Capabilities:** Following Hövelmanns et al. [48], we examine adversaries who can conduct polynomial-time computations, including quantum ones. These adversaries also have classical access to all the listed oracles, whether online or offline. In addition, as described in Section 2.2 [53], our adversaries could access quantum information from any random oracles, even if they are offline.

An adversary's capabilities are determined by their access to oracles, which enable them to communicate with protocol instances operated by users; as described in [68], the functionalities of these oracles are crucial for analyzing the protocol's security. (Please refer [78] for details). It is important to note that the Corrupt oracle enables the adversary to obtain the long-term secrets of corrupted users. Regarding our case, secrets we are referring to are the private keys that are utilized for authentication in the underlying Saber.2AKE protocols.

- $Send(U_i, s_i, \mathcal{M})$: oracle allows the adversary to send messages $\mathcal{M}$ to specific user instances $\prod_{i}^{s_i}$ and return a reply that is generated by that instance. In case "$\boldsymbol{A}$" queries that oracle with a not used instance $\prod_{i}^{s_i}$ and $\mathcal{M} \subseteq \mathcal{P}$ the $\mathbf{used_i^{s_i}}$ flag is set, $\mathbf{pid_i^{s_i}}$ initializes with $\mathbf{pid_i^{s_i}} \coloneqq \{U_i\} \cup \mathcal{M}$, and initially generated protocol message $\prod_{i}^{s_i}$ is returned.
- $Execute(\{\prod_{u_1}^{s_{u_1}}, \ldots, \prod_{u_\mu}^{s_{u_\mu}}\})$: It runs a complete protocol between the users' designated unused instances. "$\boldsymbol{\mathcal{A}}$" An adversary can access a record of every

communication sent and received via the network. A query executed by Execute Oracle is anticipated to lead to passive eavesdropping.

- $Reveal(U_i, s_i)$: This results in the value stored in $\mathbf{sk_i^{s_i}}$.
- $Test(U_i, s_i)$: Let b a bit that is selected uniformly at random. Providing that session key defined (i-e, $\mathbf{acc_i^{s_i}} = \text{true}$ and $\mathbf{sk_i^{s_i}} \neq \text{NULL}$) and instance $\prod_i^{s_i}$ is fresh (see definition) , "$\mathcal{A}$" can execute this racle query at any time when being activated. Next, the sesion key $\mathbf{sk_i^s}$ is returned when $b = 0$, while a uniformly chosen random session key is returned when $b = 1$n this model, the adversary "$\mathcal{A}$" can make aran bitrary number of test queries. However, once test oracle returns value from an instance $\prod_i^{s_i}$, this will return exact value form every instance partnered with $\prod_i^{s_i}$ (refer to description of **"Partnering"** right below).
- Corrupt (U_i): It reveals each long-term secrets from user U_i—specifically, the private keys utilized to perform authentication within "Saber'.2AKE" protocol.

4. Ensuring Correctness, Integrity, and secrecy:

To establish the security and correctness objectives, we've adopted "partnering" to indicate which instances of shared protocol sessions are associated [11].

Partnering [78]: We refer to instances $\prod_i^{s_i}$ and $\prod_j^{s_j}$ as being partnered if: $\text{sid}_\text{i}^{\text{s}_\text{i}} = \text{sid}_\text{j}^{\text{s}_\text{j}}$, $\text{sk}_\text{i}^{\text{s}_\text{i}} = \text{sk}_\text{j}^{\text{s}_\text{j}}$, $\text{pid}_\text{i}^{\text{s}_\text{i}} = \text{pid}_\text{j}^{\text{s}_\text{j}}$ and $\text{acc}_\text{i}^{\text{s}_\text{i}} = \text{acc}_\text{j}^{\text{s}_\text{j}} = true$.

If there have been no deviations from the protocol specification, then it is assumed that an instance $\prod_i^{s_i}$ has accepted the session key constructed at the end of the corresponding protocol run. In addition, in the absence of adversarial interference, all users participating in each session are expected to generate the identical session key.

Definition 3.16: *Correctness: The group key exchange protocol P considered as correct,* when in the existence of passive adversary A (i.e., A must not use Corrupt or the Send oracle) the following conditions hold: ∀ *i, j* with both "$\mathrm{acc}_i^{s_i} = \mathrm{acc}_j^{s_j} = true$ and $\mathrm{sid}_i^{s_i} = \mathrm{sid}_j^{s_j}$, we have $\mathrm{pid}_i^{s_i} = \mathrm{pid}_j^{s_j}$ and $\mathrm{sk}_i^{s_i} = \mathrm{sk}_j^{s_j} \neq \mathrm{NULL}$."

Regardless of whether adversaries are actively involved in a particular execution, a certain level of correctness should be ensured; the concept of integrity, introduced in [57], encapsulates this notion.

Definition 3.17: *key integrity:* A correct *group key exchange protocol P* is considered to have key integrity when it is highly likely that all users who have accepted with the same session identifier ($\mathrm{sid}_j^{s_j}$) possess identical session keys ($\mathrm{sk}_j^{s_j}$) and identical partner identifiers ($\mathrm{pid}_j^{s_j}$).

Now, to provide a thorough security definition, we need to clearly define the conditions under which a Test-query can be executed.

Definition 3.18: *Freshness:* The "Test-query" is just permitted for the instances hold key that can't easily be accessible to an adversary for trivial purposes. To achieve this, instance $\prod_i^{s_i}$ considered fresh when none of these following holds: –

- *For some* $U_j \in pid_i^{s_i}$*, a query Corrupt (*U_j*) was executed before a query from* $Send(U_k, s_k, \mathcal{M})$ *has taken place, for some message (or set of identities)* $\mathcal{M}$ *and some* $U_k \in pid_i^{s_i}$.
- *The adversary previously queried* $Reveal(U_j, s_j)$*: with* $\prod_i^{s_i}$ *and* $\prod_j^{s_j}$ *being partnering.*

The basic idea behind this definition is that revealing session key from instance $\prod_{i}^{s_i}$ *trivially generates session key of all instances partnered with* $\prod_{i}^{s_i}$ *these types of "attack" will be eliminated in security definition.*

Definition 3.19: *key secrecy.* For a *group key exchange protocol* to be considered secure, it is essential to limit the adversary's potential advantage. The advantage $Adv_{\mathcal{A}}(\ell)$ of a probabilistic polynomial-time adversary $\mathcal{A}$ in compromising protocol *P* is a function of security parameter and defined as follow:

$$Adv_{\mathcal{A}} := |2 \cdot Succ - 1|.$$

In this case, "Succ" represents the probability that the adversary makes queries Test solely on fresh instances and accurately predicts the bit b used by the Test oracle without breaching the freshness conditions of the instances queried with Test.

Definition 3.20: An *authenticated group key establishment protocol P* is considered secure if, for all probabilistic polynomial-time (ppt) adversary $\mathcal{A}$, the following inequality holds for some negligible function negl:

$$\mathrm{Adv}_{\mathcal{A}}(\ell) \leq \mathrm{negl}(\ell),$$

3.6 Summary

This chapter comprehensively analyzes the mathematical foundations underlying the Saber-GAKE protocol and its resilience against quantum attacks. This section highlights the essential role of lattice-based problems, including LWE, LWR, and Mod-LWR, which are fundamental to post-quantum cryptography. The problems establish a foundational framework that ensures resilience against classical and quantum adversaries, thereby enhancing the protocol's robustness in the face of quantum computing challenges to traditional cryptographic systems.

This chapter examines the influence of quantum computation on cryptographic protocols. This chapter explores quantum mechanics, specifically the concept of qubits, to elucidate how quantum capabilities may undermine current cryptographic systems. The introduction of the QROM is crucial for assessing security against quantum adversaries. QROM provides a framework for modelling quantum attacks, essential for evaluating the resilience of cryptographic protocols against quantum computational capabilities.

The chapter further elucidates the transition from IND-CPA PKE to IND-CCA KEM. This shift is essential for improving the security of encryption systems, particularly in their ability to protect against sophisticated adversaries capable of performing chosen ciphertext attacks (CCA). Enhancing the resistance of encryption schemes, these transformations strengthen the protocol's defence against various sophisticated attack strategies.

The chapter highlights the practical applications of theoretical concepts by illustrating the adaptation of two-party key exchange protocols for secure group communication. This example is relevant for dynamic and distributed IoT environments, where secure group communication is essential. The Saber-GAKE protocol utilises the discussed mathematical principles to maintain the security of communication systems.

This chapter establishes a robust theoretical framework for comprehending the cryptographic principles underlying the Saber-GAKE protocol. This work provides an in-depth analysis of the application of these principles concerning quantum threats, the robustness of encryption schemes, and their practical implementations in secure group communication within contemporary IoT networks.

CHAPTER 4

POST-QUANTUM GROUP AUTHENTICATED KEY EXCHANGE PROTOCOL

In this chapter, we describe the detailed design of the proposed GAKE protocol named "Saber-GAKE" (Figure 4.1), secure under QROM, utilizing the Saber family of post-quantum cryptographic tools. The protocol has been designed to satisfy the critical need for strong cryptographic solutions to protect communications from quantum attacks. In 2022, CRYSTALS-KYBER was declared the new standard for PQC by the NIST. During that process, SABER emerged as one of the finalist KEM algorithms. Based on the NIST report, CRYSTALS-KYBER was chosen as the standard because of the extensive study conducted on its underlying hard problem, MLWE, compared to SABER's hard problem, MLWR. However, SABER gained recognition for its lack of security, performance, or efficiency issues, making it a suitable algorithm for the post-quantum era.

The primary objective of this protocol design is to develop a GAKE solution that is resistant to quantum assaults and guarantees mutual authentication, confidentiality, and integrity of communications within a group. The protocol utilizes Abdalla's compiler design to transform a two-party key exchange into a group key exchange, ensuring efficiency and security in resource-constrained contexts like IoT networks.

This chapter will explore the specific design considerations and a detailed overview of the protocol construction. This includes a description of the SABER KEM and other important cryptographic primitives. Subsequently, we will address the design considerations that influenced the protocol's development, focusing on scalability, efficiency, and security. This chapter's primary objective is to explain the protocol extensively, which will be separated into key generation, Encryption, and decryption

phases. Algorithms and pseudocode will accompany each phase to elucidate the steps involved.

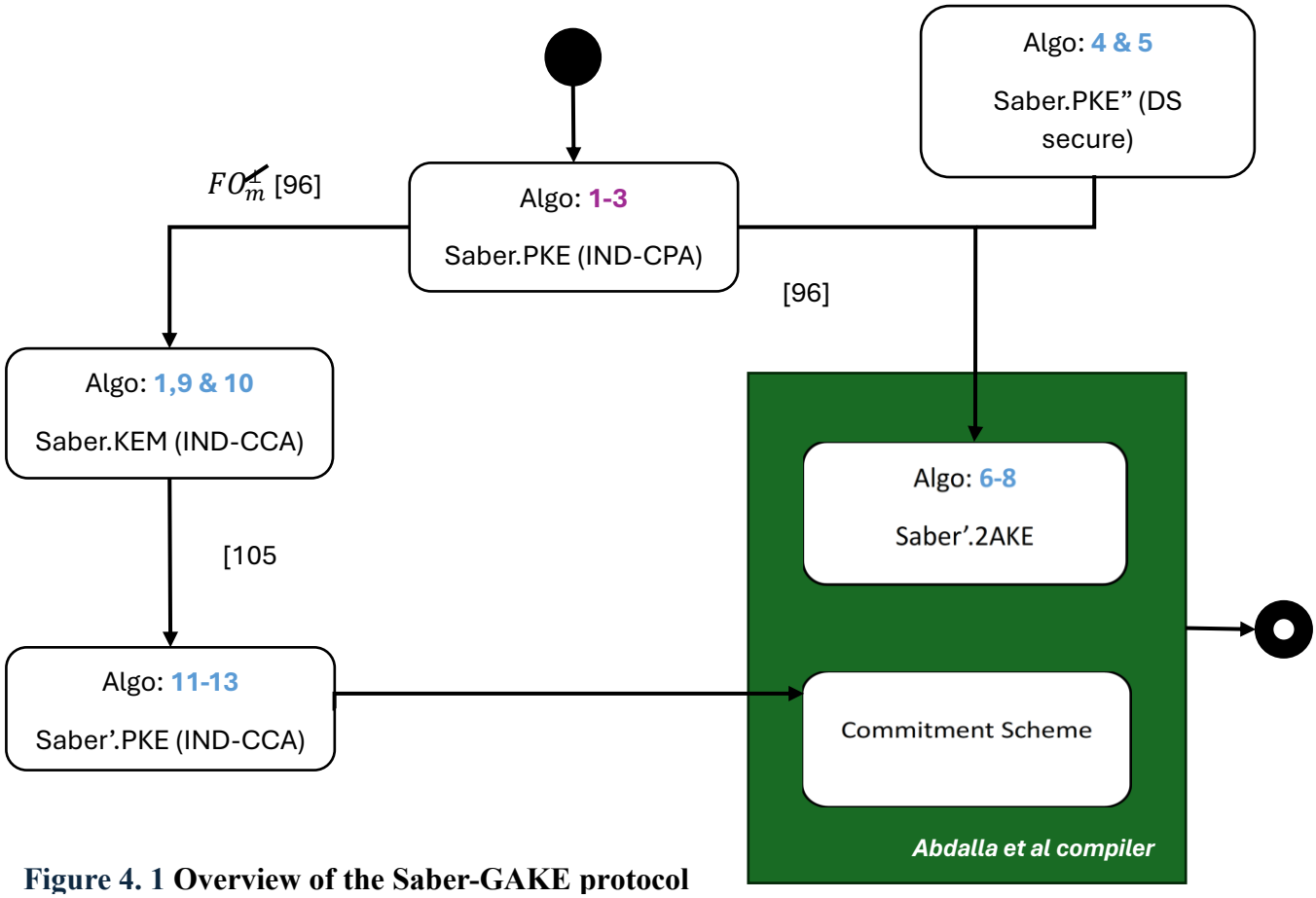

Figure 4. 1 Overview of the Saber-GAKE protocol

4.1 Saber.PKE (CPA-secure):

In this subsection, we discuss a PKE scheme called Saber.PKE that is CPA-secure and was introduced in [29]. The Saber.PKE is a variant of the Saber.KE key exchange protocol, as illustrated in Figure 4.1 and described in [94]. Firstly, let's begin by discussing the definition required to construct the Saber.PKE. The Saber.KE scheme can be transformed into a public-key encryption scheme called Saber.PKE, which provides CPA security. This transformation is described in [94]. The message space $M \in \{0,1\}^n$ and message $m \in M$ is represented by elements in R_q with coefficients {0, 1}. Algorithms 1-3 describe the public-key encryption scheme Saber.PKE = (KeyGen,

Enc, Dec), with the setup parameters matching those of the previously mentioned key-exchange scheme. When you include the optional parameter r in your call to Saber.ENC, it will be used as a seed to generate the

secret vector $\boldsymbol{s'}$.

	Alice		Bob
1.	$\boldsymbol{seed_A} \leftarrow \boldsymbol{U}(\{\mathbf{0},\mathbf{1}\}^{\mathbf{256}})$		
2.	$\boldsymbol{A} \leftarrow \boldsymbol{gen}(seed_A) \in R_q^{lxl}$		
3.	$\boldsymbol{s} \leftarrow \beta\mu(R_q^{lx1})$		$\boldsymbol{s'} \leftarrow \beta\mu(R_q^{lx1})$
4.	$\boldsymbol{b} = bits(\boldsymbol{As} + \boldsymbol{h}, \epsilon_q, \epsilon_p) \in R_p^{lx1}$		$\boldsymbol{A} \leftarrow gen(seed_A) \in \boldsymbol{R_q^{lxl}}$
5.		$\xrightarrow{\boldsymbol{b}, seed_A}$	$\boldsymbol{b'} = bits(A^T s' + h, \boldsymbol{\epsilon_q}, \boldsymbol{\epsilon_p}) \in R_p^{Lx1}$
6.			$v' = b^T bits(s', \boldsymbol{\epsilon_p}, \boldsymbol{\epsilon_p}) + h_1 \in R_p$
7.	$\boldsymbol{v} = \boldsymbol{b'^T}\, \boldsymbol{bits}(\boldsymbol{s}, \boldsymbol{\epsilon_p}, \boldsymbol{\epsilon_p}) + \boldsymbol{h_1} \in \boldsymbol{R_p}$		$c = bits(v', \boldsymbol{\epsilon_p} - 1, \boldsymbol{\epsilon_t}) \in R_t$
8.	$k = \text{bits}(v - 2^{\boldsymbol{\epsilon_p}-\boldsymbol{\epsilon_t}-\mathbf{1}}c + \boldsymbol{h_2}, \boldsymbol{\epsilon_p}, \mathbf{1})$	$\xleftarrow{\boldsymbol{b'}, c}$	$\boldsymbol{k'} = bits(v', \boldsymbol{\epsilon_p}, 1)$
9.	$key_{Alice} = kdf\ (k)$		$key_{Bob} = kdf\ (k')$

Figure 4. 2 Protocol 2: Saber.KE key exchange[94]

Theorem 4.1 [94]**:** For any adversary $\mathcal{A}$ against Saber.PKE, there exists an adversary B against Saber.KE such that $Adv_{saber.PKE}^{ind-cpa}(\mathcal{A}) = Adv_{saber.KE}^{ind-rnd}(\mathcal{B})$. Furthermore, Saber.PKE is $(1-\delta)$ correct if and only if Saber.KE is $(1-\delta)$ correct.

Proof: The proof proceeds by illustrating the similarity between Saber.PKE and a combination of Saber.KE with a one-time pad of the message m with k'.KE. In Saber.KE, it is worth noting that the most significant bit of each coefficient of v' corresponds to the corresponding (pre)key bits of k'. Therefore, in line 5 of Protocol 2, the addition function acts as a one-time pad of the message bits m, incorporating the coefficients of the (pre)key k' into the key exchange scheme (Protocol 2). Therefore, it can be argued that the level of security provided by our encryption is proportional to the level of security offered by our key exchange mechanism, given that the parameters are same.

Likewise, it is evident that the Saber.PKE algorithm yields accurate results when the keys k and k' are similar. Consequently, the accuracy of the encryption method is equivalent to the accuracy of the key exchange in Protocol 2.

Saber.PKE (CPA secure)

Algorithm 1: $\boldsymbol{Saber.KeyGen}()$	Algorithm 2: $Saber.Enc(pk = (b, seed_A), m \in M; r)$
1. $\boldsymbol{seed_A} \leftarrow \boldsymbol{U}(\{\boldsymbol{0,1}\}^{\boldsymbol{256}})$	1. $A \leftarrow gen(seed_A) \in R_q^{lxl}$
2. $\boldsymbol{A} \leftarrow \boldsymbol{gen}(\boldsymbol{seed_A}) \in \boldsymbol{R_q^{lxl}}$	2. $s' \leftarrow \beta\mu(R_q^{lx1})$
3. $\boldsymbol{s} \leftarrow \boldsymbol{\beta}\mu(\boldsymbol{R_q^{lx1}})$	3. $b' = bits(A^T s' + h, \epsilon_q, \epsilon_p) \in R_p^{Lx1}$
4. $\boldsymbol{b} = \boldsymbol{bits}(\boldsymbol{As} + \boldsymbol{h}, \boldsymbol{\epsilon_q}, \boldsymbol{\epsilon_p}) \in \boldsymbol{R_p^{Lx1}}$	4. $v' = b^T bits(s', \epsilon_p, \epsilon_p) + h_1 \in R_p$
5. $\boldsymbol{return}$ $(\boldsymbol{pk} := (\boldsymbol{b}, \boldsymbol{seed_A}), \boldsymbol{sk} := \boldsymbol{s})$	5. $c_m = bits(v' + 2^{\epsilon_p - 1} m, \epsilon_p, \epsilon_t + 1) \in R_{2t}$ 6. $return\ c: = (c_m, b')$

Algorithm 3: $\boldsymbol{Saber.Dec}(\boldsymbol{sk} = \boldsymbol{s}, \boldsymbol{c_m}, \boldsymbol{b'})$

1. $\boldsymbol{v} = \boldsymbol{b'^T}\ \boldsymbol{bits}(\boldsymbol{s}, \boldsymbol{\epsilon_p}, \boldsymbol{\epsilon_p}) + \boldsymbol{h_1} \in \boldsymbol{R_p}$
2. $\boldsymbol{m'} = \boldsymbol{bits}(\boldsymbol{v} - \boldsymbol{2^{\epsilon_p - \epsilon_t - 1} c_m} + \boldsymbol{h_2}, \boldsymbol{\epsilon_p}, \boldsymbol{1}) \in \boldsymbol{R_2}$
3. $\boldsymbol{return}\ \boldsymbol{m'}$

4.2 FROM Saber.PKE to Saber'.AKE: The FO_{AKE} Transformation

The FO_{AKE} Transformation, proposed in [95], is a general construction that can convert any IND-CPA secure scheme into AKE (Authenticated Key Exchange), which is likely to be secure under QROM. The result achieved through the FO_{AKE} transformation is incredibly effective when it comes to communication. In their study, Hövelmanns et al [95] have proposed a security model along with two security notions for two-message

AKEs. These notions include key indistinguishability against active attacks (IND-AA) and a slightly weaker notion of indistinguishability against active attacks without state reveal in the test session (IND-StAA). We prefer the second option because the security of the AKE obtained through the FO_{AKE} transformation is proven under a slightly weaker model. Nevertheless, the provided information meets our requirements since it is detailed in Section 5 of the mentioned paper [96] (the extended version of [95]). This concept, known as IND-StAA, ensures the required level of security in the compiler discussed in [78].

The IND-StAA model can be described at a high level. It is mentioned that the session key remains indistinguishable from a random one, regardless of the situation.

- The intruder has either the long-term secret key or the secret state information (but not both) of both parties involved in the test session if it hasn't tampered with the received message.
- If the message received during the test session is tampered with by an intruder, they would be unable to gain access to the long-term secret key of the test session's peer or the current state of the test session.

The security of the FO_{AKE} transformation has been successfully demonstrated in the QROM, if the PKE satisfies the IND-CPA condition. In addition, they have demonstrated the ability to create counterfeit ciphertexts that closely resemble authentic encryptions, with a minimal chance of the sampling algorithm generating a genuine encryption. This idea is referred to as Disjoint Simulatability (DS) of ciphertexts, and it is defined in [95] as follows:

4.2.1 Transformation of Secure Saber.PKE (IND.CPA) to Saber.PKE" (DS secure):

The TPunc ("Puncturing and Encrypt-with-Hash") modulization transformation, as described in [97], demonstrates the ability to convert any IND-CPA secure PKE scheme into a deterministic DS. The author [96] demonstrates this transformation in two simple steps.

By puncturing the message space at a specific message and generating fake encryptions, the Punc transformation could convert any IND-CPA secure public-key encryption scheme into a DS secure one.

Transformation T[98] can transform a probabilistic public-key encryption scheme into a deterministic one. The transformed scheme is DS, since PKE is also DS and IND-CPA secure.

The Construction. Consider an encryption scheme $PKE = (KG, Enc, Dec)$with a message space $\mathcal{M}$ and a randomness space $\mathcal{R}$. Let's consider PKE with an additional sampling algorithm $\overline{Enc}$ (see Definition 3.). We define the algorithms 1, 4 & 5 of $Saber.PKE'' = (KG' := Saber.KeyGen(), Saber.Enc''(pk, m),$

$Saber.Dec''(sk, c), \overline{Enc}' := \overline{Enc})$ and associate it with $Saber.PKE$ and random oracle $G : \mathcal{M} \rightarrow \mathcal{R}$ as $Saber.PKE" = T[Saber.PKE, G]$. It is important to note that $Saber.Enc''$ computes the ciphertext in a deterministic manner, specifically as $c := Saber.Enc(pk, m; G(m))$.

The lemma 3.1[96] demonstrates the implication of combining IND-CPA and DS security of PKE on the DS security of PKE′.

Saber.PKE" (IND-CPA + DS)

Algorithm 4: $\boldsymbol{Saber.Enc''(pk, m)}$	Algorithm 5: $Saber.Dec''(sk, c)$

01 $\boldsymbol{c := Saber.Enc(pk, m; G(m))}$	01 $m' := Saber.Dec(sk, c)$
02 $\boldsymbol{return\ c}$	02 $if\ m' = \perp\ or\ Saber.Enc(pk, m'; G(m')) = c$
	03 $return\ \perp$
	04 $else\ return\ m'$

After applying the TPunc transformation to Saber.PKE, we have successfully achieved a secure PKE scheme against both IND-CPA and DS attacks. This scheme is known as Saber.PKE. The theorem and proof mentioned in the paper provide strong evidence that Saber.PKE" meets the security requirements, as another theorem from a different source explains. Now, we can use it to create a two-party AKE that meets IND-StAA security in the QROM. The resulting scheme, known as Saber'.2AKE, as algorithm 6-8. Here, G and H are random oracles, while H'_R, H'_{L1}, H'_{L2}, and H'_{L3} are internal random oracles that cannot be accessed directly and could be implemented with a pseudorandom function.

Saber'.2AKE

Algorithm 6: $\boldsymbol{Init(sk_i, pk_j)}$

1. $\boldsymbol{m_j \xleftarrow{\$} M}$
2. $\boldsymbol{c_j := Saber.Enc''(pk_j, m_j;\ G(m_j))}$
3. $\boldsymbol{(\widetilde{pk},\ \widetilde{sk}) \leftarrow Saber.KeyGen()}$
4. $\boldsymbol{M := (\widetilde{pk}, c_j)}$
5. $\boldsymbol{st := (\widetilde{sk}, m_j, M)}$
6. $\boldsymbol{return\ (M, st)}$

Algorithm 7: $\boldsymbol{Der_{resp}(sk_j, pk_i, M)}$:

Algorithm 8: $Der_{init}(sk_i, pk_j, M', st)$:

1. $(\widetilde{\boldsymbol{pk}}, \boldsymbol{c_j}) := \boldsymbol{M}$
2. $\boldsymbol{m_i}, \widetilde{\boldsymbol{m}} \xleftarrow{\$} \boldsymbol{M}$
3. $\boldsymbol{c_i} := \boldsymbol{Saber.Enc''}(\boldsymbol{pk_i}, \boldsymbol{m_i}; \boldsymbol{G}(\boldsymbol{m_i}))$
4. $\tilde{\boldsymbol{c}} := \boldsymbol{Saber.Enc''}(\widetilde{\boldsymbol{pk}}, \widetilde{\boldsymbol{m}}; \boldsymbol{G}(\widetilde{\boldsymbol{m}}))$
5. $\boldsymbol{M'} := (\boldsymbol{c_i}, \tilde{\boldsymbol{c}})$
6. $\boldsymbol{m'}_j := \boldsymbol{Saber.Dec''}(\boldsymbol{sk_j}, \boldsymbol{c_j})$
7. $\boldsymbol{if}\ \boldsymbol{m'}_j = \perp\ \boldsymbol{or}\ \boldsymbol{c_j} \neq \boldsymbol{Saber.Enc''}(\boldsymbol{pk_i}, \boldsymbol{m_i}; \boldsymbol{G}(\boldsymbol{m_i}))$
8. $\boldsymbol{K'} := \boldsymbol{H'}_{\boldsymbol{R}}(\boldsymbol{m_i}, , \boldsymbol{c_j}, \widetilde{\boldsymbol{m}}, \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{M}, \boldsymbol{M'})$
9. $\boldsymbol{else}$
10. $\boldsymbol{K'} := \boldsymbol{H}(\boldsymbol{m_i}, \boldsymbol{m'_j}, \widetilde{\boldsymbol{m}}, \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{M}, \boldsymbol{M'})$
11. $\boldsymbol{return}\ (\boldsymbol{M'}, \boldsymbol{K'})$

1. $(c_i, \tilde{c}) := M'$
2. $(\widetilde{sk}, m_j, M := (\widetilde{pk}, c_j)) := st$
3. $m'_i := Saber.Dec''(sk_i, c_i)$
4. $\widetilde{m'} := Saber.Dec''(\widetilde{sk}, \tilde{c})$
5. $if\ m'_i = \perp\ or\ c_i \neq Saber.Enc''(pk_i, m_i'; G(m_i'))$
6. $if\ \widetilde{m'} = \perp$
7. $K := H'_{L1}(c_i, m_j, \tilde{c}, i, j, M, M')$
8. $else$
9. $K := H'_{L2}(c_i, m_j, \widetilde{m'}, i, j, M, M')$
10. $else\ if\ \widetilde{m'} = \perp$
11. $K := H'_{L3}(m'_i, m_j, \tilde{c}, i, j, M, M')$
12. $else\ K := H(m'_i, m_j, \widetilde{m}, i, j, M, M')$
13. $return\ K$

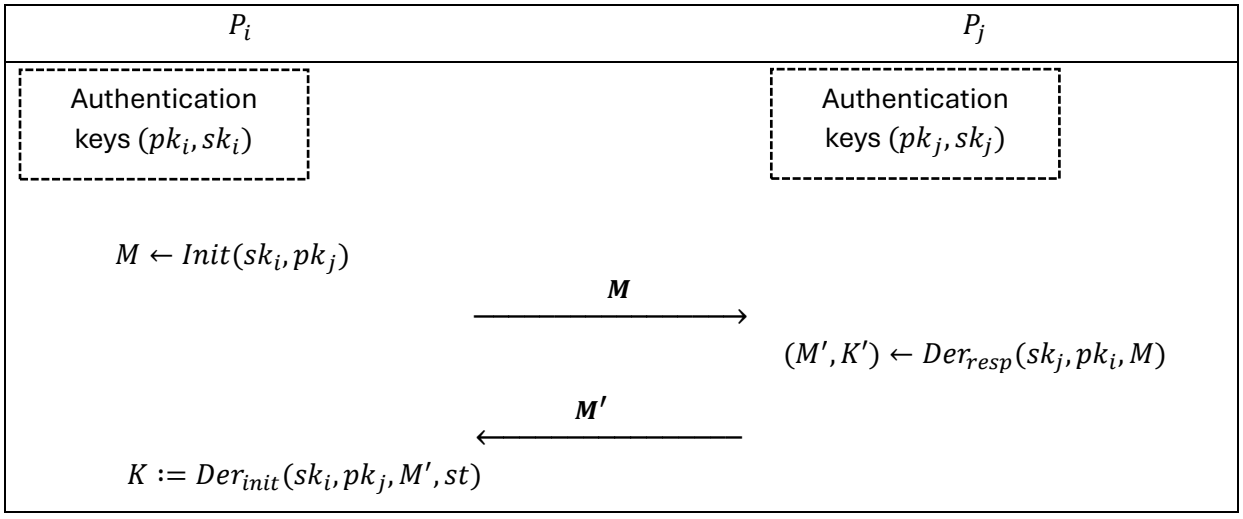

Figure 4. 3 Saber'.2AKE secure under QROM

4.3 Commitment Scheme

In this section, we delve into the development of our commitment scheme, which was previously mentioned in section 3.3. Commitment schemes allow participants to hide a value until later in the protocol securely. Ensuring that all participants begin with identical initial values or inputs is crucial for maintaining the protocol's consistency. Having a commitment scheme that is non-interactive, non-malleable, and perfectly binding while also achieving non-malleability for multiple commitments is of utmost importance. We outline our commitment scheme, which consists of two steps:

1) Apply $FO_m^{\perp}$ transformation [95] to convert Saber.PKE (IND-CPA PKE) to Saber′.KEM (IND-CCA KEM).
2) Applying the transformation proposed in [99] to convert Saber'.KEM (IND-CCA KEM) to Saber'.PKE (IND-CCA PKE).

To achieve an IND-CCA secure KEM with Saber.PKE, we make use of the FO_m transformation. This is like the FO_{AKE} transformation, which transforms a PKE scheme that provides both IND-CPA and DS security into a CCA-secure KEM. As shown in [95], the FO_m transformation distinguishes itself from other transformations by its ability to handle correctness errors effectively and provide stronger security reduction. If the PKE is not already DS, this requirement can be waived without any noticeable impact on efficiency. With Saber.PKE, efficiency is not compromised. It achieves both IND-CPA security and, as shown in Theorem 2, DS security as well. Algorithms 1, 9, and 10 showcase the KEM Saber′ = (Saber.PKE.KeyGen, Encaps, Decaps) achieved through the application of the FO_m transformation to $Saber'.PKE$. Here, G and H are random oracles, and H_r is an internal random oracle that cannot be accessed directly and could be implemented with a pseudorandom function.

After applying the transformation introduced in [99] to Saber, a secure one-time symmetric key encapsulation (SKE or DEM) obtains an IND-CCA PKE. This scheme is referred to as $Saber'.PKE$ (Algo 11-13). Theorem 5 in [62] supports this transformation's security. As mentioned in [78], obtaining a commitment scheme with the necessary security properties is possible only from the IND-CCA PKE.

Saber'.KEM (IND-CCA KEM)

Algorithm 9: $\boldsymbol{Saber'.Encaps(pk)}$	Algorithm 10: $Saber'.Decaps(sk, c)$
1. $\boldsymbol{m \xleftarrow{\$} M}$	5. $m' = Saber.Dec''(sk, c)$
2. $\boldsymbol{c := Saber.Enc''(pk, m;\ G(m))}$	6. $if\ m' = \perp\ \ or\ Saber.Enc''\big(pk, m'; G(m')\big) \neq c$
3. $\boldsymbol{k := H(m)}$	7. $return\ k := H_r(c)$
4. $\boldsymbol{return\ (k, c)}$	8. $else$
	9. $return\ k := H(m')$

Saber'.PKE (IND CCA)

Algorithm 11: $\boldsymbol{Saber'.PKE.KeyGen(\,)}$	Algorithm 12: $Saber'.PKE.Enc(pk, m)$
1. $\boldsymbol{(pk, sk) \leftarrow Saber.KeyGen(\,)}$	3. $(c, K) \leftarrow\ Saber'.Encaps(pk)$
2. $\boldsymbol{return\ (pk, sk)}$	4. $c' := E(K, m)$
	5. $return\ c" = (c, c')$

Algorithm 13: $\boldsymbol{Saber'.PKE.Dec(sk, c" = (c, c'))}$

6. $\boldsymbol{K := Saber'.Decaps(sk, c)}$
7. $\boldsymbol{return\ m := D(K, c')}$

4.4 SABER-GAKE Protocol Description:

The proposed GAKE Protocol (as shown in Fig. 4.4) in this research relies on two essential primitives: One of the finalists in the NIST standardization process

competition is Saber KEM [94]. The other is the Abdalla et al compiler [78], which facilitates the conversion of an Authenticated Key Exchange (AKE) between two parties into a Group Authenticated Key Exchange (GAKE). It considers that users be structured cyclically, allowing each user to be addressed using a publicly accessible and consistent index throughout all executions. The GAKE can be built using fundamental tools to demonstrate its security in the QROM, which is a more resilient security model compared to the commonly used ROM. Furthermore, the process of key generation is considered contributively, indicating that no individual user can ascertain the output session key.

Users consider themselves as transparent entities that can be represented as polynomial-time probabilistic Turing machines, indicating their absence of quantum computing capability. These users are limited to exchanging communications across an unsecured network that adversaries completely control. This means that enemies could add, delay, delete, or resend messages as they see fit. Moreover, the adversaries have computational powers beyond those of the users. It is assumed that the adversaries can perform quantum calculations in polynomial time and possess quantum access to any hash function, which is modeled as a random oracle.

4.4.1 From 2-AKE to GAKE protocol

The compiler attributes developed by Abdalla et al. allow for transforming an arbitrary 2AKE into a GAKE protocol in section 3.2. This section discusses our compiled construction using a compiler [78]. We propose a GAKE protocol with a strong emphasis on security against the quantum adversary. It utilizes a post-quantum 2AKE protocol and compiler (discussed in Section 3.3) with certain modifications. Figure 4.5 depicts the protocol.

Simplified Key Derivation:

The author of [86] uses a slightly revised version of the compiler from [78] by simplifying the process of computing the session key and identifier, which we adhere to. Initially, a complex procedure was used to extract these values from the shared master key (K). Instead, two hash functions are utilized:

- Generate the session key (sk_i) from the master key (K).
- Generate the session identifier (sid_i) from the master key (K).

This simplification is justified because the underlying 2AKE is already secure in the QROM, which assumes idealized hash functions. Given the security assumption, there is no need for the complicated key extraction process designed for non-idealized hash functions. We can remove Tools 1 & 2 from Section 3.3.2 with depend only on $\hat{H}$ and $\hat{F}$.

Ensuring Security: Besides the simplified key derivation, [86] also considers an extra security requirement mentioned by Nam in [100]. Theorem 1 of [78] depends on the 2AKE protocol ensuring message integrity, guaranteeing that messages remain unaltered throughout the protocol execution.

The original 2AKE may not have explicitly ensured integrity. Thus, [86] suggests making a small adjustment to the two-party 2AKE to meet this requirement and avoid a particular replay attack mentioned in [90] (Theorem 1 [68] proves implicit that integrity is maintained; see argument related to Game 1 [78]). Modifications to the message construction or verification steps within the 2AKE protocol might be necessary for this tuning.

Round 1-2: : Execution: Each participant U_i (where $i\ ranges\ from\ 0\ to\ n-1$), perform "$Saber'.2AKE$" operation with U_i and $U_{[i+1]}$ where $[k] = k\ mod\ n$ (i.e., all indices should be taken cyclically). Where two rounds as follows:

Round 1: At completion, each $U_i \in \mathcal{G} = \{U_0, U_1, \ldots, U_{n-1}\}$ maintains two pairwise keys $\vec{K}_i$ and $\overleftarrow{K}_i$ shared with $U_{[i+1]}$.

Round 2: At completion, each $U_i \in \mathcal{G} = \{U_0, U_1, \ldots, U_{n-1}\}$ maintains two pairwise keys $\vec{K}_i$ and $\overleftarrow{K}_i$ shared with $U_{[i-1]}$,respectively. Note that, $for\ i = 0, \ldots, n-1$, $\vec{K}_i = \overleftarrow{K}_{[i+1]}$.

Round 3:

- Calculation: Each U_i computes $X_i = \vec{K}_i \oplus \overleftarrow{K}_i$,
  and selects a random r_i to derive commitment "$C_i = Saber'.PKE(i, X_i; r_i)$ ",
- Broadcast: Each U_i telecasts the message:

$$M_i^1 = (U_i, C_i)$$

Round 4:

- Broadcast: Each U_i telecasts the message:

$$M_i^2 = (U_i, X_i, r_i)$$

- Verification: Each U_i verifies the condition $X_0 \oplus X_1 \oplus X_2 \ldots \oplus X_{n-1} = 0$ and ensures the correctness of the commitments. If any check fails, U_i terminates protocol execution setting $acci: = FALSE$. If all check pass, U_i $acc_i := TRUE$ and $pid_i := \mathcal{G}$.
- Computation: Each U_i computes $(n-1)$ values:
  $K_{i-j} := \overleftarrow{K}_i \oplus X_{i-1} \oplus \ldots \oplus X_{i-j}$, for $j = 1,2, \ldots, n-1$.
  Next, U_i defines a master key K as:

$$K = \left(\overleftarrow{K}_0, \overleftarrow{K}_1 \ldots \ldots, \overleftarrow{K}_{n-1}, \mathcal{G}\right)$$

  and sets the session key and session identifier as

$$sk_i: = \hat{H}(K) \text{ and } sid_i := \hat{F}(K)$$

where $\hat{H} : \{0,1\}^* \rightarrow \{0,1\}^\ell$ and $\hat{F} : \{0,1\}^* \rightarrow \{0,1\}^\ell$ are hash functions and $\ell \in \mathbb{N}$ is the security parameter.

Figure 4. 4 Proposed Group key Exchange Protocol

By integrating simplified key derivation and enforcing integrity modifications to the underlying 2-AKE, our GAKE construction provides a proven secure solution against quantum adversaries.

4.4.2 Correctness of Saber.GAKE

Theorem 4.2: The GAKE protocol presented in Figure 4.5 using Saber'.2AKE is correct in the presence of passive adversary, ensuring that all the honest participants derive the same non-null session key $\mathrm{sk}_i^{s_i} = \mathrm{sk}_j^{s_j} \neq \mathrm{NULL}$ and share the same participant identifier $\mathrm{pid}_i^{s_i} = \mathrm{pid}_j^{s_j}$ when they accept the session key.

Assumption 1: Let v and w are two different integers. Considering these integers.

- If $|| \mathrm{v} - \mathrm{w} ||_\infty < {}^{p}/_{4(1-{}^{1}/_{t})}$, reconciliation between parties will successful.
- If $|| \mathrm{v} - \mathrm{w} ||_\infty > {}^{p}/_{4(1+{}^{1}/_{t})}$, reconciliation between parties will be unsuccessful.

Assumption 2: In LWR-based cryptosystems, it is necessary to introduce a discrete uniformly distributed error to determine the upper bound of the error probability distribution. This error value $e_r \in Z_p$ expressed as ${}^{-p}/_{4} < || e_r||_\infty < {}^{p}/_{4}$.

Proof: Using Reconciliation conditions

- Each participant U_i generate an initial message M and state st using **Algorithm 6**. This includes generating a random message m_j, computing its ciphertext c_j, and generating a new key pair $(\widetilde{pk}, \widetilde{sk})$.
- The initial message $M := (\widetilde{pk}, c_j)$is sent to the other participant U_j.
- Upon receiving M, U_j generates its messages m_i, $\widetilde{m}$ and corresponding ciphertext c_i, $\tilde{c}$ using **Algorithm 7**.
- U_j decodes the received ciphertext c_j to obtain m'_j and verifies it is against m_j.
- If the reconciliation conditions hold, U_j computes the session key K'.

- The initiator U_i receives $(c_i, \tilde{c})$ and decodes them using **Algorithm 8.**
- If the reconciliation conditions hold, U_i computes the session key K.
- The reconciliation condition guarantees that any discrepancies between the exchanged values are rectified if assumptions 1 and 2 hold.
- In Round 3, each participant computes local values X_i and exchanges commitments C_i.
- The session key K is derived using a hash function over the combined values, ensuring consistency if reconciliation holds.
- In Round 4, each participant verifies that XOR of all X_i values equal zero, and commitments are correct. If all checks pass, the session key K is accepted, and the session is marked successful ($\mathrm{acc}_i^{s_i} = \mathrm{acc}_j^{s_j} = true$).
- The protocol ensures that $\mathrm{sk}_i^{s_i} = \mathrm{sk}_j^{s_j} \neq \mathrm{NULL}$ and $\mathrm{pid}_i^{s_i} = \mathrm{pid}_j^{s_j}$ when both participants accept the session key, ensuring correctness by **Definition 3.16**.

4.5 Security Arguments and Proofs

Our compiled GAKE relies on the security properties of its underlying tools:

- **Saber′.2AKE (Figure 4.4):** This two-party AKE protocol is secure in the sense of IND-StAA (implying security in the sense essential for the original compiler from [78] as explained in section 5 [95]) and can be modified to achieve integrity (Definition 4 [86]).
- **Saber′. PKE:** This encryption scheme's IND-CCA security (Section: Commitment Scheme [78]) enables the development of a non-interactive commitment scheme that is both non-malleable and perfectly binding for several commitments.

Security Justification: The combination of secure building blocks underpins our compiled GAKE's security:

- **IND-StAA in Saber′.2AKE:** Ensures the security of derived session keys by thwarting adversaries from using protocol messages to gain an advantage.
- **Integrity in Modified Saber′.2AKE:** The primary concept involves incorporating a secondary random oracle F, which will be applied to the same input as H during the key derivation process to produce a session identifier. Subsequently, it is straightforward to assert the integrity of this modified Saber′.2AKE construction is achieved in both ROM and QROM because of the collision resistance of random oracles involved (see Section 3.2.2). Well, let's consider the scenario where k is equal to k. Given the nature of H and F as random oracles, it can be confidently stated that their collision resistance ensures that both participants will almost certainly have identical partner identifiers and, consequently, utilize the same session key k. This argument holds true in both the classical and quantum-accessible random oracle model (see Section 3.2). In the sequel, we assume that this modification has been implemented, resulting in the attainment of integrity by Saber′.2AKE.
- **IND-CCA in Saber.PKE:** Enables construction of a secure commitment scheme, crucial for key confirmation and preventing key-related attacks within the compiled GAKE.

4.5.1 Security of Our Proposed Saber-GAKE:

Theorem 4.3: The protocol presented in Figure 4.5 is a correct and secure GAKE protocol that fulfills integrity, as defined in Definitions 3.16, 3.17, and 3.20 in the random oracle model.

Proof: This proof is a relatively straightforward adaptation of the security proof of Theorem 1 of [78], which we use as primary tool in our construction.

Correctness: During an honest implementation of the protocol, all participants will conclude by agreeing upon and computing the identical session identity and session key. This guarantees that all parties reach an agreement in the outcome of the protocol.

Integrity: All the oracles that accept same session identifiers also have a significant chance of holding the same master key K and pid (which could be derived from K) due to the collision-resistance property of the random oracle $\widehat{F}$. As an outcome, they will also generate the identical session key, $\widehat{H}(K)$.

Key secrecy: Proof of key secrecy will be carried out by a sequence of games, starting with an actual attempt to compromise the key secrecy of the Saber-GAKE and ending with a game in which the adversary's advantage is zero and the difference in the adversary's advantage between any two consecutive games can be limited. The advantage of the adversary $\mathcal{A}$ in G_i is denoted by $Adv(\mathcal{A}, G_i)$ in accordance with standard notation. Additionally, to enhance clarity, we categorize the Send queries into three groups, which are determined by the protocol stage with which the query is associated. These categories are as follows: Send-1..., Send-4. Send query linked with round t for t = 1, 2, ...,4 is denoted by Send-t.

Game 0 (Real Attack): The adversary $\mathcal{A}$ interacts with the real protocol where all parameters and secrets are taken as in actual scheme.

The first game in a sequence implies to a real attack, in which long-term secret associated with each user and all the public parameters in common reference string are identical as actual scheme.

$\boldsymbol{G_0}$

1. $\boldsymbol{Initialize\ public\ parameters\ and\ long-term\ secrets.}$
2. $\boldsymbol{\mathcal{A}\ interacts\ with\ the\ protocol\ with\ various\ oracles.}$
3. return $\boldsymbol{Adv(\mathcal{A},\ G_0) = Adv(\mathcal{A})}$

Game 1 (Random key Replacement): For $i = 1,2,\ldots,n$, we modify the Send and Execute oracles, as if instance $\prod_{i}^{s_i}$ still consider fresh at the end of Round 2, the keys $\overleftarrow{K_i}$ and $\overrightarrow{K_i}$ which shares with instances $\prod_{i-1}^{s_{i-1}}$ and $\prod_{i+1}^{s_{i+1}}$ are replaced with the random values .

- If no query Corrupt U_j has been asked by adversary from some $U_j \in \mathbf{pid_i^{s_i}}$ before a $Send(U_k, s_k, \mathcal{M})$ query for some $U_k \in \mathbf{pid_i^{s_i}}$, then consider the instance $\prod_{\boldsymbol{i}}^{\boldsymbol{s_i}}$ as fresh at the end of Round 2.
- The probability of an adversary breaking the security of any underlying 2AKE protocol is the limitation of the difference between this game and G_0, implies as

$$\boldsymbol{|Adv(\mathcal{A}, G_1) - Adv(\mathcal{A}, G_0)| \leq 2 \cdot Adv_{2-\mathrm{AKE}}(\ell, 2 \cdot \mathbf{q}_{send}),}$$

Construct an adversary $\boldsymbol{\mathcal{A}_{2-AKE}}$

- $\boldsymbol{\mathcal{A}_{2-AKE}}$ is constructed from a given $\boldsymbol{\mathcal{A}}$ distinguishing G_1 from G_0.
- $\boldsymbol{\mathcal{A}_{2-AKE}}$ is provided access to a simulation of the Saber'.2AKE protocol.
- It associates each user instance $\prod_{i}^{s_i}$ in the GAKE protocol with two independent instances of the same user in the 2AKE protocol.

Handling queries by $\boldsymbol{\mathcal{A}_{2-AKE}}$

- Whenever, $\boldsymbol{\mathcal{A}}$ makes a corrupt query then $\boldsymbol{\mathcal{A}_{2-AKE}}$ respond using Corrupt (U_i) oracle of 2AKE protocol and return the same value.
- For answering Execute queries, $\boldsymbol{\mathcal{A}_{2-AKE}}$ queries the **Execute** oracle of the 2AKE protocol with the corresponding instances to obtain the transcript for Round 2.
- To simulate the subsequent rounds, $\boldsymbol{\mathcal{A}_{2-AKE}}$ queries Test oracle of "2-AKE protocol" with corresponding instance and uses the returned values as keys $\overleftarrow{K_i}$ and $\vec{K_i}$**.**
- $\boldsymbol{\mathcal{A}_{2-AKE}}$ queries the Send oracle of the 2AKE protocol with the corresponding instance for Send 1-2 queries and returns its response.
- For Send queries connect to round 3 and 4, $\boldsymbol{\mathcal{A}_{2-AKE}}$ set the values of the keys $\overleftarrow{K_i}$ and $\vec{K_i}$ by querying either **Test** or **Reveal** oracle of the 2AKE protocol with the corresponding instance and proceed with the simulations. More specifically if the instance $\prod_{i}^{s_i}$ in group protocol is still fresh at starting of Round 3, $\boldsymbol{\mathcal{A}_{2-AKE}}$ query **Test** oracle, otherwise Reveal oracle.

Distinguishing game states

- $\boldsymbol{\mathcal{A}_{2-AKE}}$ is successful in distinguishing between G_0 and G_1 by determining whether the Test oracle returns a random element from the key space or reveals the actual exchanged key.

  Game 2 (Commitment Acceptance Modification): In this game, we modify the Simulation of **Send** oracle so that fresh instance $\prod_{i}^{s_i}$ doesn't accept in Round 4 whenever one commitment C_j for $j \neq i$ it receives in Round 3 was generated by the simulator but not by the respective instance $\prod_{j}^{s_j}$, $j \neq i$ in identical session.

  - An adversary A can distinguish between G_1 and G_2 if it replays a commitment that should have been accepted in Round 3 in G_1

- Given that X_i is a random value, probability of A distinguishing the games is negligible.

$$|Adv(\mathcal{A}, G_2) - Adv(\mathcal{A}, G_1)| \leq negl(\ell)$$

To prove this, consider a session with instances $\prod_1^{s_1}, \prod_2^{s_2}, \ldots, \prod_n^{s_n}$. Each instance $\prod_i^{s_i}$ expects commitments C_j corresponding to values X_j such that: $X_1 \oplus X_2 \ldots \oplus X_n = 0$

$$\text{results, } \overleftarrow{K}_1 \oplus \vec{K}_1 \oplus, \ldots\ldots\ldots \oplus, \overleftarrow{K}_n \oplus \vec{K}_n$$

An instance $\prod_i^{s_i}$, $X_i = \overleftarrow{K}_i \oplus \vec{K}_i$ where, $\overleftarrow{K}_i$ is the key shared with U_{i-1} and $\vec{K}_i$ with U_{i+1}.

The commitment C_j includes the index of user U_j and is perfectly binding, ensuring that the commitments cannot be tampered with by any adversary. The adversary A is unable to accurately reveal the commitments if they are rearranged among the session participants due to the binding nature of the commitments.

Given that, the keys $\overleftarrow{K}_i$ and $\vec{K}_i$ are random, probability of any XOR sum of keys not consisting exactly of the keys in one session (thus not cancelling out to zero) being zero is: $1/2^k$.

An adversary $\mathcal{A}$ can perform at most q_{send} attempts to distinguish between the games, every attempt has an $1/2^k$ of producing an incorrect commitment sum that the instance would still accept. So, the total probability that adversary A can disguising between G_2 and G_1 is $q_{send}/2^k$.

Game 3 (Adversary-generated Commitment Modification): This game we include the commitment generated by adversary. We modify the Simulation of **Send** oracle so that fresh instance $\prod_{i}^{s_i}$ doesn't accept in Round 4 whenever one commitment C_j for $j \neq i$ it receives in Round 3 was generated by adversary.

- The adversary's advantage in the current game differs only insignificantly from the previous game.

$$|Adv(\mathcal{A}, G_3) - Adv(\mathcal{A}, G_2)| \leq negl(\ell)$$

To prove, we will construct an attacker $\mathcal{A}_{com}$ against the non-malleability of the commitment scheme from an adversary A that can distinguish between G_3 from G_2.

Construction of $\boldsymbol{\mathcal{A}_{com}}$

- $\boldsymbol{\mathcal{A}_{com}}$ receives commitments $C_i = Saber'.PKE(i, X_i; r_i)$, for $i = 1,2, \ldots\ldots, n$ where X_i values are random bitstrings that fulfil $X_1 \oplus X_2 \ldots \oplus X_n = 0$. For bitstring X_i' , the 2n-ary relation is given by $\mathcal{R}(X_1, \ldots X_n, X_1', \ldots X_n') = 1$

if and only if,

$X_1' \oplus X_2' \ldots \oplus X_n' = 0$ and $X_i = X_i'$ for at least one index $i \in \{1, \ldots\ldots, n\}$

- $\boldsymbol{\mathcal{A}_{com}}$ simulates the environment for $\mathcal{A}$, for session not involving $\prod_{i}^{s_i}$, $\boldsymbol{\mathcal{A}_{com}}$ answer the queries as G_2.
- On the other hand, for session involving $\prod_{i}^{s_i}$, $\boldsymbol{\mathcal{A}_{com}}$ uses commitment C_i to answer Send-3 queries. When $\mathcal{A}$ provides $\prod_{i}^{s_i}$ with set of commitments C'_j for $j \neq i$, $\boldsymbol{\mathcal{A}_{com}}$ halts and outputs the set of commitments along with C_i.

Success Probability:

- $\boldsymbol{\mathcal{A}_{com}}$ will be successful if it accurately identifies the instance $\prod_{i}^{s_i}$ of receiving a collection of commitments that includes at least one commitment generated by an adversary that passes the verification test.
- So far, the simulation by $\boldsymbol{\mathcal{A}_{com}}$ is perfect, making Games G3 and G2 completely identical.

Non-malleability:

- The commitment scheme's non-malleability ensures that $\boldsymbol{\mathcal{A}_{com}}$ success probability is only slightly higher than that of an adversary who lacks knowledge of the commitments C_i.
- In the absence of visibility into the commitments, the adversary's probability of generating valid commitments C_i, where $X_1' \oplus X_2' \dots \oplus X_n' = 0$ and, is $q_{send}/2^k$ as previous game.

Game 4 (Random Session keys): In this game, the simulation of the Send & Execute oracle modified when session key compute. The simulator contains assignments list: $(K_1, \dots\dots, K_n, \mathcal{G})$. Once an instance receives final Send-4 query, simulator checks if a master key corresponding to list $(K_1, \dots\dots, K_n, \mathcal{G})$ already exists. If master key exist it is assigned to the instance otherwise a session key $\mathrm{sk}_{\mathrm{i}}^{\mathrm{Si}} \in \{0,1\}^{\ell}$ is chosen uniformly at random.

- Key idea is that however, if even messages from Round 4 transmitted, the master key still retains a satisfactory amount of entropy. Consequently, the random oracle $\widehat{H}$ produces an output that is practically impossible to differentiate from a randomly generated $\mathrm{sk}_{\mathrm{i}}^{\mathrm{Si}}$ with a negligible probability.

$$|Adv(\mathcal{A}, G_4) - Adv(\mathcal{A}, G_3)| \leq negl(\ell)$$

By illustrating that the adversary's ability to distinguish between Game 4 and Game 3 is negligible, we strengthen the overall argument that in Game 4, session keys are chosen uniformly at random, and the adversary does not possess any significant advantage. Therefore, adversary has no advantage in Game 4.

$$Adv(A, G_4) = 0$$

Theorem 4.4: The presented protocol in Figure 4.5 is a correct and secure GAKE protocol that fulfills integrity, as defined in Definitions 3.16, 3.17, and 3.20, in the Quantum random oracle model.

Proof (sketch): The proof follows the same structure as Theorem 3, with particular attention to quantum-accessible random oracles. We ensure that arguments from G_4 remains valid in this context.

In Game 4, the simulator changes the Execute and Send oracles while establishing the session key. The simulator stores a collection of strings $(K_1, \ldots\ldots, K_n, \mathcal{G})$. Upon receiving the final Send-4 query, the system computes $K_1, \ldots., K_n$. and verifies whether a corresponding master key has been issued. If a master key is present, it is allocated to the instance. Otherwise, a session key $\mathrm{sk}_{\mathrm{i}}^{\mathrm{s_i}}$ is selected randomly and uniformly from the set $\{0,1\}^k$.

The keys are selected uniformly at random and are not known to adversary. An adversary may simply perceive this alteration if it has already requested the exact key strings $(K_1, \ldots\ldots, K_n, \mathcal{G})$ to the quantum random oracle $\widehat{H}$. In the quantum random oracle model, the probability of the adversary having queried the exact key string

$(K_1, \ldots\ldots, K_n, \mathcal{G})$ is negligible. Given the large key space, the chances of the adversary correctly guessing the exact string required for $\widehat{H}$ is quite relatively low.

Considering the negligible probability of the adversary correctly guessing the key string, the output of the random oracle $\widehat{H}$ is indistinguishable from a randomly generated session key $sk_i^{s_i}$. This suggests that an adversary is unable to differentiate between session key generated from master key and one that is uniformly random, thus we have:

$$|Adv(A, G_4) - Adv(A, G_3)| \leq negl(\ell)$$

Given that the session keys in Game 4 are chosen uniformly random, the adversary does not possess any advantage in distinguishing them. Therefore, the Adversary has no advantage in Game 4.

$$Adv(A, G_4) = 0$$

4.6 ROM-secure Primitives:

In this section, we discuss the ROM primitives we used to compare with "Compiled-Kyber" ROM primitives.

Table 4.1 Comparison between parameter sets used for ROM

	Security Level	Ciphertext Length	Public key length	Secret key Length
Compiled-Kyber[73]	128	736	800	1632
	192	1088	1184	2400
	256	1568	1568	3168
	128	736	**672**	**1568**

Saber-	192	1088	**992**	**2304**
GAKE	256	**1472**	**1312**	**3040**

As IND-CCA KEM original version of Saber used as demonstrate as (Algo 20-22) in [101], form the foundation of our ROM secure implementation. The two-party authenticated key exchange (AKE) protocol referred as Saber.2AKE illustrate in Figure 4.5. In this case parties are assumed to be A and B incorporate the following algorithms:

$InitA\ (Algorithm\ A.1),\ sharedB\ (Algorithm\ A.2),\ sharedA\ (Algorithm\ A.3).$

Just as in QROM, the commitment scheme is derived from an "IND-CCA PKE scheme". Specifically, Algo: 11-13 are used with Saber instead of Saber'.

Saber.2AKE secure under ROM

ALGORITHM A.1 $initA(pk_j)$

1. $(pk, sk) \leftarrow Saber.KeyGen()$
2. $(c_j, K_j) \leftarrow Saber.Encaps(pk_j)$
3. $return\ (M := (pk, c_j), sk, K_j)$

ALGORITHM A.2

$sharedB(pk, c_j, pk_i, sk_j)$

1. $(C, K) \leftarrow Saber.Encaps(pk)$
2. $(c_i, K_i) \leftarrow Saber.Encaps(pk_i)$
3. $K'_j := Saber.Decaps(sk_j, c_j)$
4. $K := H(K, K_i, K'_j)$
5. $return\ (K, M' := (c, c_i))$

ALGORITHM A.3 $sharedA(sk, ski, c, ci, Kj)$

1. $K' := Saber.Decaps(sk, c)$
2. $K'_i := Saber.Decaps(sk_i, c_i)$
3. $K := H(K', K'i, Kj)$
4. $return\ K$

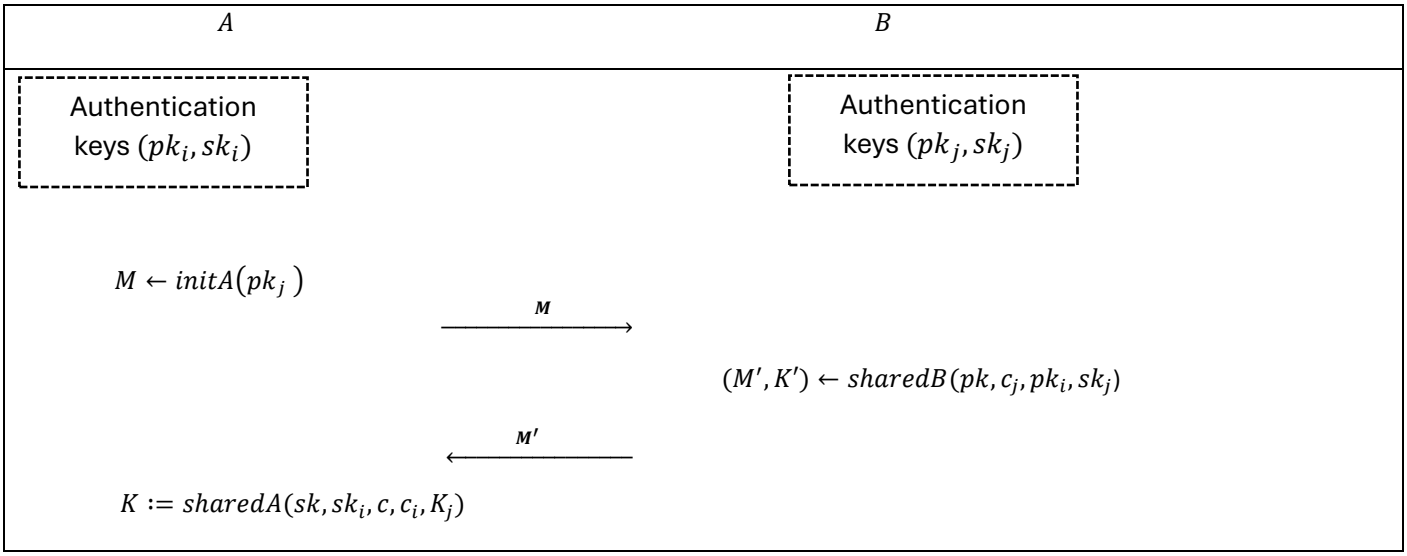

Figure 4. 5 Saber.2AKE secure under ROM

4.7 Novelty and Comparative Analysis with Compiled Kyber

The proposed Saber-GAKE protocol introduces several key innovations over existing post-quantum GAKE constructions, particularly when compared to the Compiled Kyber GAKE protocol by Pablos et al. While both schemes aim to achieve secure group key exchange in a post-quantum setting without relying on expensive signature mechanisms, Saber-GAKE differentiates itself through its design choices, performance focus, and suitability for resource-constrained IoT environments.

Firstly, Saber-GAKE is built upon the Module-LWR assumption with the Saber KEM, whereas Compiled Kyber relies on the Module-LWE assumption via the Kyber KEM. The Module-LWR foundation offers inherent advantages for IoT devices, including smaller ciphertexts and reduced computational overhead due to its efficient rounding techniques. This makes Saber-GAKE more suitable for constrained devices where memory and processing resources are limited. Additionally, Saber-GAKE is designed explicitly for scalability, being evaluated for large groups of up to 2000 participants, a feature critical for real-world IoT deployments but not explicitly addressed in the original Compiled Kyber work.

Moreover, the performance analysis conducted for Saber-GAKE includes a detailed theoretical estimation of execution times and memory consumption on ARM Cortex-M series processors, showcasing its viability for deployment in practical IoT settings. Although the performance evaluation for IoT environments is based on existing Saber benchmark implementations rather than direct hardware deployment, it nonetheless provides strong evidence of the protocol's efficiency. In contrast, the Compiled Kyber implementation focuses primarily on general system performance without specific optimization for embedded or low-power platforms. Furthermore, Saber-GAKE provides a complete security proof in the Quantum Random Oracle Model (QROM), maintaining rigorous cryptographic assurances while optimizing for lightweight operation. These distinctions collectively underline the novelty and significant contribution of the Saber-GAKE protocol within the domain of post-quantum secure group communication, particularly for IoT applications.

4.8 Summary

This chapter delves into the design and theoretical basis of the Saber-GAKE protocol, a quantum-resistant solution for secure group communication in IoT applications. The chapter presents the protocol's cryptographic principles, namely the MLWR problem, which is the foundation for the Saber KEM. The Saber-GAKE protocol builds on this basis to handle the specific issues of secure group communication in resource-constrained IoT networks.

The chapter also describes using the Abdalla et al. compiler to convert two-party authenticated key exchange (AKE) into a group authenticated key exchange (GAKE) protocol. This structure provides strong cryptographic security against quantum attackers, supported by rigorous proofs using the QROM. Furthermore, the protocol

includes streamlined key generation and session integrity procedures, improving theoretical robustness while maintaining practical feasibility.

This chapter describes the Saber-GAKE protocol's potential to provide secure communication in a variety of IoT applications, with an emphasis on scalability, security, and efficiency. It offers the framework for understanding the protocol's novel approach to addressing the key difficulties of post-quantum cryptography, establishing its theoretical relevance and significance in the rapidly evolving field of IoT security.

CHAPTER 5

EXPERIMENTAL SETUP AND RESULTS

5.1 Experiment Results: Proposed "Saber-GAKE" vs "Compiled Kyber"

This section compares the experiment's results with those of the reference implementation. This experiment assesses the effectiveness of our proposed group key exchange protocol, "Saber-GAKE," built upon the "Saber KEM" algorithm. We compare its performance with a GAKE protocol called "Compiled Kyber," built on the "Kyber KEM" algorithm. We utilize the "Compiled Kyber" reference implementation found at https://github.com/jiep/kyber-gake for comparison, referred to as "ref".

To provide context for the comparison, Table 5.1 summarizes the differences in the security levels targeted by the two protocols. Table 5.2 details the parameter sets used for each KEM, highlighting their structural and computational variations. Additionally, Table 5.3 outlines the properties of different variants of Saber as implemented in the original Saber KEM specification [94], which forms the basis for parameter selection in our Saber-GAKE protocol.

The following subsections present the comparative experimental results, discussing execution time, memory usage, and communication overhead for both Saber-GAKE and Compiled Kyber across various operating environments.

Table 5. 1 Comparison w.r.t Security level

Security Level		
LightSABER ≈ AES-128	AES-128 ≈ **Kyber-512**	**LightSABER** ≈ **Kyber-512**
SABER ≈ AES-192	AES-192 ≈ **Kyber-768**	**SABER** ≈ **Kyber-768**

FireSABER ≈ AES-256	AES-256 ≈ **Kyber-1024**	**FireSABER** ≈ **Kyber-1024**

Table 5. 2 Comparison between parameter sets

	Security Level	k	n	μ, η	q	p
Compiled-Kyber[73]	128	2	256	x	3329	x
	192	3	256	x	3329	x
	256	4	256	x	3329	x
Saber-GAKE	128	2	256	10	8192	1024
	192	3	256	8	8192	1024
	256	4	256	6	8192	1024

k: Module Dimension **q, p:** Modulo Values **μ, η:** Distribution Parameter

Table 5. 3 Properties of each variant of Saber implemented in [94]

	Variants	Ciphertext	Public key	Secret key	Poly coin	Poly	Poly vector	Poly compress	Poly vector compress	Scales KEM
Compiled-Kyber[73]	Kyber-512	736	800	736	x	384	768	96	640	x
	Kyber-768	1088	1184	1152	x	384	1152	128	960	x
	Kyber-1024	1568	1568	1536	x	384	1536	160	1408	x
Saber-GAKE	LightSABER	736	672	832	320	416	832	320	640	96
	SABER	1088	992	1248	256	416	1248	320	960	128
	FireSABER	1472	1312	1664	192	416	1664	320	1280	192

5.1.1 Experimental Environment Setup:

The experiments were performed deploying Visual Studio Code on a machine with an 11th Generation Intel I Core I i7-1165G7 processor operating on Ubuntu 22.04.3.49.0, with 16 GB of RAM (Table 5.4). The binaries for various implementations were constructed using Cmake, utilizing the "-DMAKE_BUILD_TYPE=Release" option and gcc as the C compiler. The binaries were linked statically. The "Ref" implementation was compiled using the gcc compiler with the settings "-O3 -fwrapv".

5.1.2 Ref (Saber-GAKE) vs ref (Compiled Kyber) implementation:

Tables 5.5 and 5.6 comprehensively analyze the performance difference between the two cryptographic systems in terms of different security levels and cryptographic operations. Table 5.5 shows that Ref (Saber-GKE) performs better than ref (Complied Kyber) in several metrics, exhibiting superior performance in ROM and QROM security models with faster speeds. It has been noted that the "Ref" implementation is generally 4 to 18% faster in ROM and 16 to 31% faster in QROM compared to the "ref" implementation. This indicates that Saber may have an edge in situations that prioritize quickness while maintaining a strong security level. On the other hand, Kyber falls behind in terms of efficiency when performing several operations quickly. Table 5.6 highlights faster execution times of Saber in all four rounds of cryptographic operations, as indicated by the total efficiency rate metric. The efficiency of Saber-GKE is especially evident in its ability to obtain cryptographic results faster than Compiled Kyber.

Table 5. 4 Hardware specifications

Features	Values
Processor	11th Gen IntelI Core I i7-1165G7 @ 2.80GHz
Operating System	Ubuntu 2204.3.49.0
RAM	16 GB

Table 5. 5 Comparison of the speed of different operations between the implementations, depending on the security level. It is shown how many times faster is the implementation Ref with respect to ref

Parameters	Security Model	KEM			Commitment		2-AKE		
		KeyGen	Encaps	Decaps	Commit	Check	Init/InitA	Der_{resp} /sharedB	Der_{init} /sharedA

LightSaber (Ref) Kyber512 (ref)	ROM	19.58%	16.02%	20.98%	19.40%	19.36%	18.27%	21.82%	21.82%
LightSaber (Ref) Kyber512 (ref)	QROM	24.56	23.19%	27.32%	23.02%	23.04%	24.25%	25.50%	30.55%
Saber (Ref) Kyber768 (ref)	ROM	2.68%	3.03%	2.13%	7.06%	6.90%	2.04%	1.15%	4.45%
Saber (Ref) Kyber768 (ref)	QOM	9.39%	5.19%	9.98%	10.21%	10.11%	9.14%	9.12%	14.67%
FireSaber (Ref) Kyber1024 (ref)	ROM	23.54%	20.06%	22.91%	26.32%	26.23%	23.13%	22.74%	24.30%
FireSaber (Ref) Kyber1024 (ref)	QROM	28.28%	26.61%	29.42%	26.94%	26.55%	26.96%	27.00%	31.14%

5.1.3 KEM, 2-AKE & Commitment Schemes:

An evaluation was carried out to compare the performance of Kyber-Saber (QROM) and Kyber-Saber (ROM) in terms of several cryptographic operations, such as Two-Party Authenticated Key Exchange (2-AKE), Key Encapsulation Mechanism (KEM), and Commitment Scheme. The results consistently indicate that Saber outperforms Kyber. Both Saber (ROM and QROM) exhibits superior efficiency compared to Kyber across all operations (KeyGen, Encaps, Decaps), as demonstrated in Figure 5.4. Saber demonstrates significant advancements. Regarding Key Generation, Encapsulation, and Decapsulation operations, Saber demonstrates better timings in ROM and QROM configurations when compared to Kyber. Furthermore, during the execution of the Two-Party AKE algorithm, Saber implementations exhibit significantly improved performance compared to Kyber, as shown in Figure 5.2. In the Commitment Scheme, Saber exhibits superior execution times in both ROM and QROM variants, further highlighting its edge in comparison over Kyber (Figure 5.1).

Saber outperforms various security and operation tiers by a substantial margin, varying from 1.15 % to 31.14%, as quantified in the comprehensive comparison in Table 5.5. The figures illustrate that the QROM variant of Saber exhibits a significantly greater performance advantage than the ROM variant. This observation highlights Saber's enhanced effectiveness and resilience in the face of quantum adversaries, rendering it especially attractive in situations where precedence is placed on quantum security.

5.1.4 Saber-GKE vs Complied Kyber:

The table 5.7 and 5.8 provides a comprehensive overview of the performance differences among Saber-GAKE and Compiled-Kyber cryptographic primitives across various security models through its comparative analysis. It is noteworthy that Round 4 is the most time-consuming round among all implementations, which greatly affects efficiency rates. A notable observation pertains to the better effectiveness rates exhibited by Saber variants compared to Kyber when managing Round 4. The performance variations are additionally underscored in Figure 5.3, where the contrasting round-wise execution times, total execution times, initialization times, and efficiency rates are graphically represented. The exceptional efficacy consistently demonstrated by Saber implementations can be attributed to their optimized execution of cryptographic operations. The significance of evaluating performance while considering both the security model and the cryptographic primitive is highlighted by these results; Saber variants have proven to be especially effective in practical cryptographic operations.

Table 5. 6 Comparison of Efficiency percentage of Ref over ref.

Parameters	Security Model	Init	Round1-2	Round3	Round4	Total time	Efficiency Rate
LightSaber (Ref)	ROM	36.104	**21.053**	**10.806**	**1887.314**	**1955.28**	

Kyber512 (ref)		**18.435**	22.145	9.1889	2250.398	2300.17	**14.99%**
LightSaber (Ref)	QROM	31.458	**15.915**	**10.334**	**1619.105**	**1676.822**	
Kyber512 (ref)		**18.029**	18.251	9.2067	1946.536	1992.030	**15.82%**
Saber (Ref)	ROM	34.596	**35.061**	**12.815**	**3046.606**	**3129.074**	
Kyber768 (ref)		**25.010**	38.796	12.865	3754.298	3830.969	**18.32%**
Saber (Ref)	QOM	33.718	25.607	12.331	2648.426	**2720.093**	
Kyber768 (ref)		**26.955**	34.164	13.313	3701.963	3776.413	**27.97%**
FireSaber (Ref)	ROM	45.749	**56.183**	**24.999**	**5316.615**	**5443.551**	
Kyber1024 (ref)		**40.086**	62.952	40.065	5560.694	5703.798	**4.56%**
FireSaber (Ref)	QROM	**35.318**	**40.693**	**22.307**	**4241.182**	**4339.515**	
Kyber1024 (ref)		39.310	51.827	44.092	5396.614	5531.840	**21.55%**

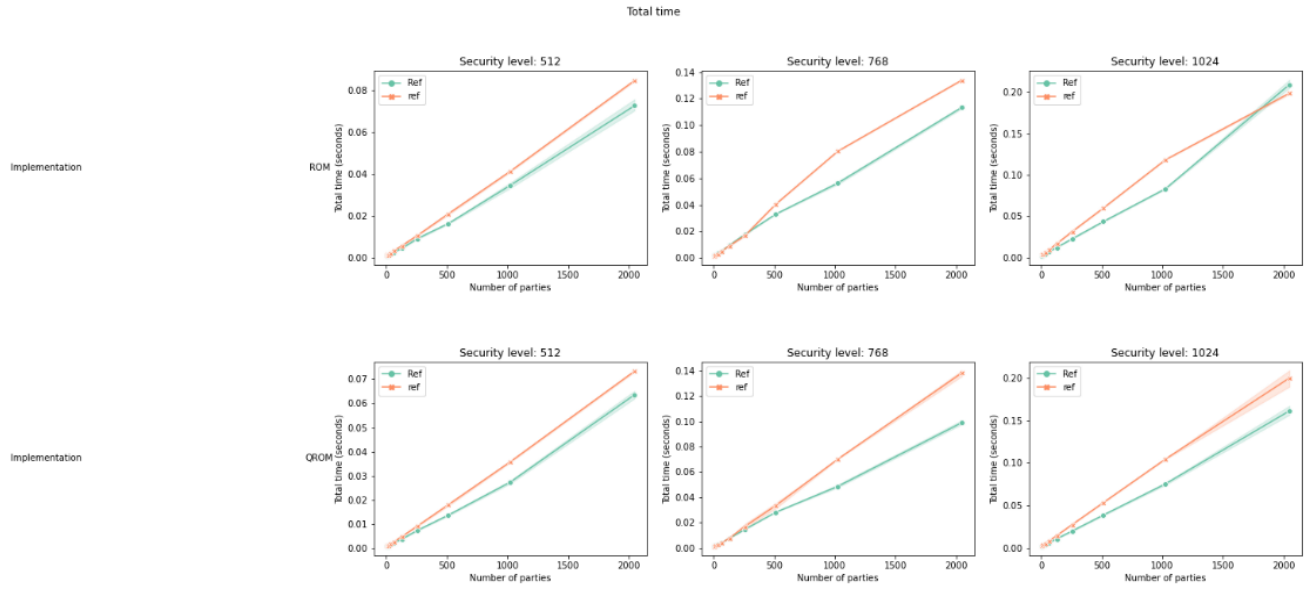

Figure 5. 1 Comparison between Total protocols time depending on the number of parties, the security level, and primitives (QROM or ROM).

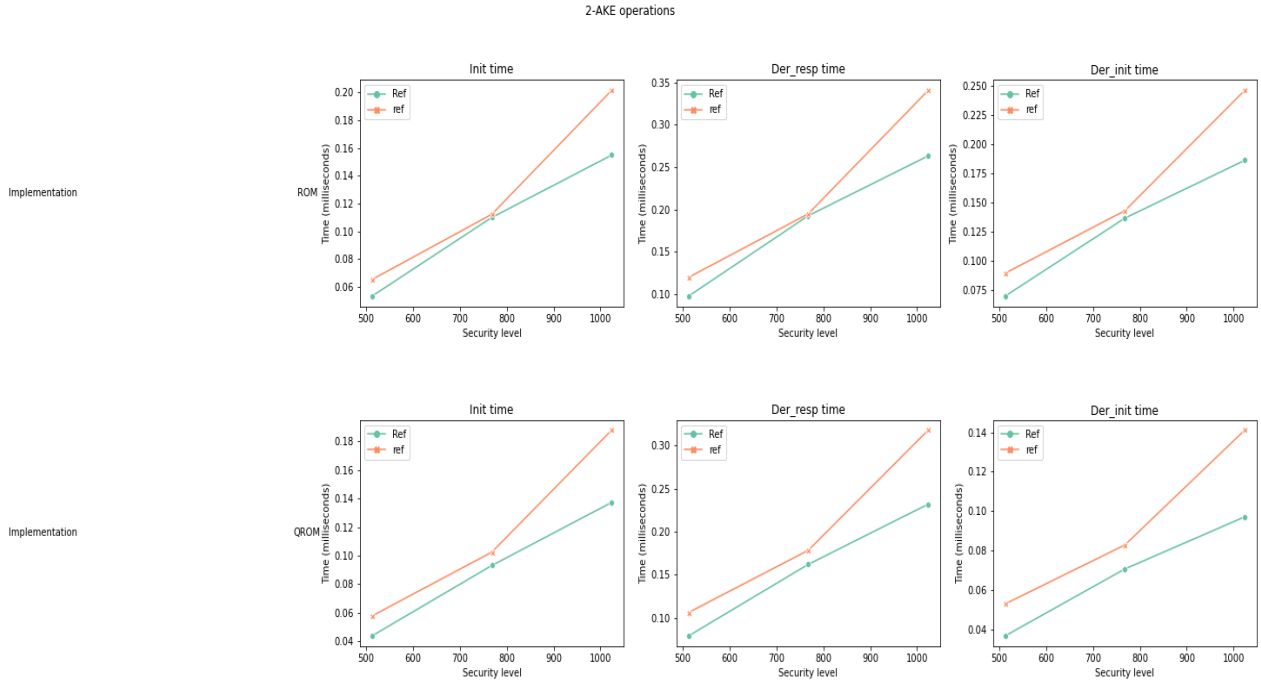

Figure 5. 2 Comparison between Runtime of 2-AKE algorithms depending on the security level and the primitives used (QROM or ROM).

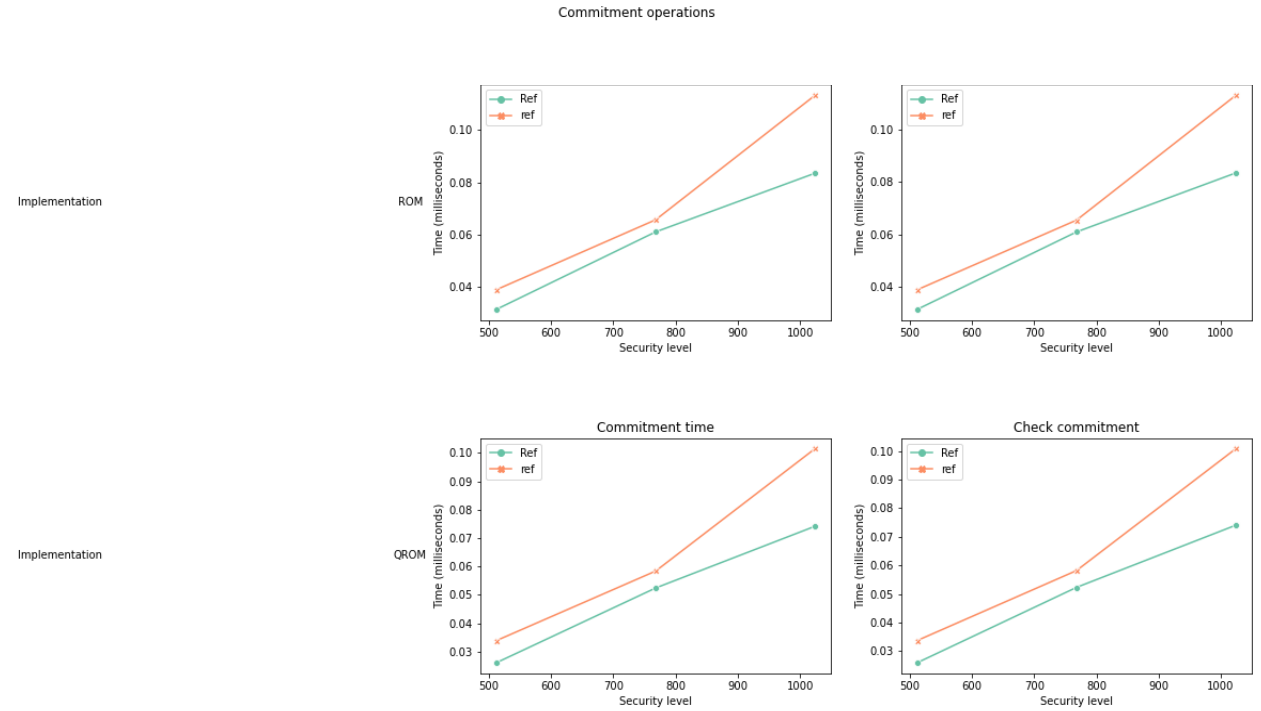

Figure 5. 3 Comparison between Runtime of commitment scheme algorithms depending on the security level and the primitives used (QROM or ROM)

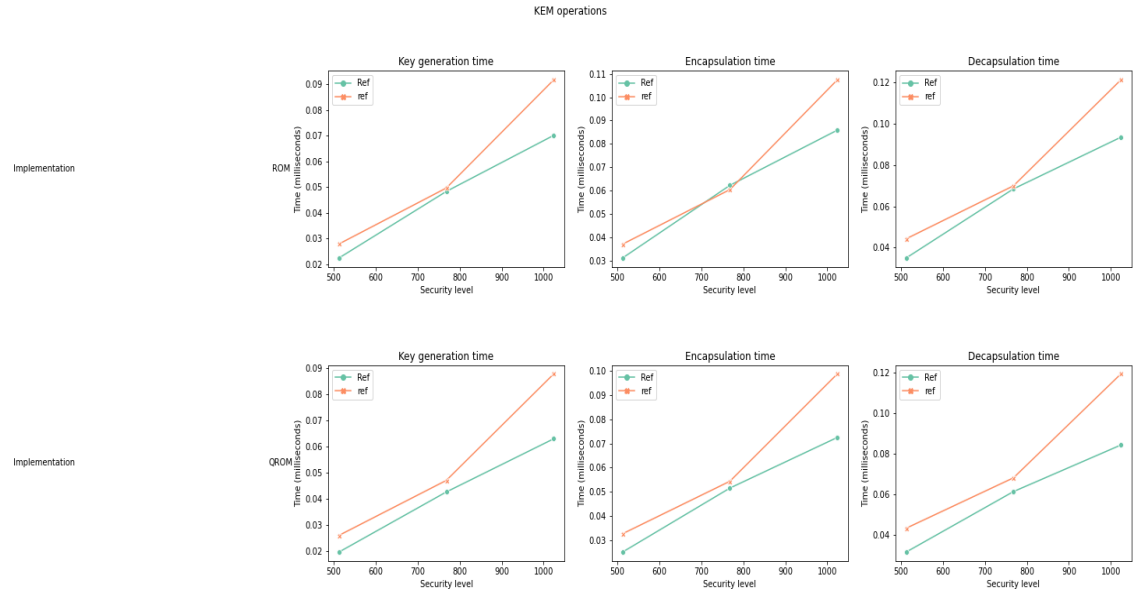

Figure 5. 4 Comparison between Runtime of KEMs operations depending on the security level and the primitives used (QROM or ROM).

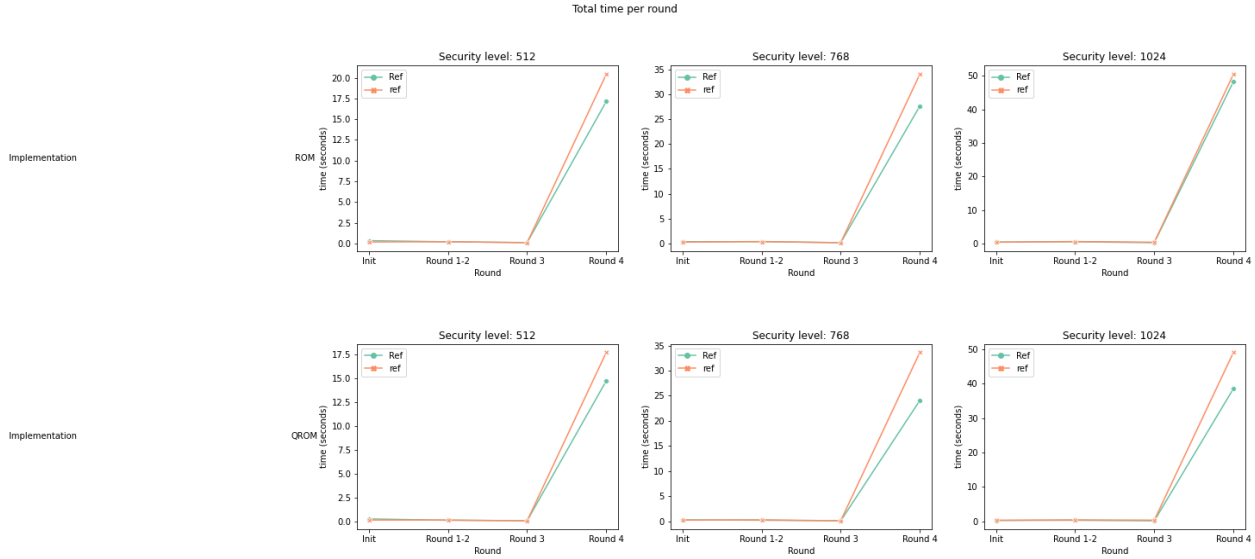

Figure 5. 5 Comparison of the Percentage of total protocol time spent in each round depending on the security level and primitives used (QROM or ROM).

Table 5. 7 Comparison of the time taken per round at various security levels based on the number of parties secured under the ROM

Parties	security	init_time (ref/Ref)	Round12 (ref/Ref)	Round3 (ref/Ref)	Round4 (ref/Ref)	total_time (ref/Ref)
2048	Kyber512 (ref) LightSaber (Ref)	13.51 /29.92	10.82/11.7	6.65/8.46	1702.12/1436.9	1733.1/1486.98
1024	Kyber512 (ref) LightSaber (Ref)	3.51 /4.79	5.67/5.01	1.59/1.58	411.92/342.17	422.69/353.55
512	Kyber512 (ref) LightSaber (Ref)	0.97 /0.98	2.83/2.09	0.53/0.42	102.7/79.89	107.03/83.38
256	Kyber512 (ref) LightSaber (Ref)	0.29/0.3	1.37/1.12	0.2/0.16	25.13/21.65	26.99/23.23
128	Kyber512 (ref) LightSaber (Ref)	0.1/0.08	0.7/0.57	0.08/0.07	6.35/4.96	7.23/5.68
64	Kyber512 (ref) LightSaber (Ref)	0.04 /0.03	0.39/0.29	0.04/0.04	1.64/1.31	2.11/1.67
32	Kyber512 (ref) LightSaber (Ref)	0.02 /0.01	0.19/0.14	0.03/0.02	0.39/0.33	0.63/0.5
16	Kyber512 (ref) LightSaber (Ref)	0.01/0.0	0.09/0.07	0.02/0.02	0.1/0.08	0.22/0.17
8	Kyber512 (ref) LightSaber (Ref)	0.0/0.0	0.04/0.03	0.01/0.01	0.02/0.02	0.07/0.06
4	Kyber512 (ref) LightSaber (Ref)	0.0/0.0	0.03/0.02	0.02/0.01	0.01/0.01	0.06/0.04
2	Kyber512 (ref) LightSaber (Ref)	0.0/0.0	0.01/0.01	0.01/0.01	0.0/0.0	0.02/0.02
2048	Kyber768 (ref) Saber (Ref)	17.39/27.0 8	17.97/17.48	8.84/8.82	2698.94/2269.0	2743.14/2322.3 8
1024	Kyber768 (ref) Saber (Ref)	5.36/5.28	10.96/8.3	2.68/2.62	803.65/560.26	822.65/576.46

512	Kyber768 (ref) Saber (Ref)	1.64/1.5	5.49/4.61	0.83/0.78	198.72/161.52	206.68/168.41
256	Kyber768 (ref) Saber (Ref)	0.4/0.48	2.15/2.33	0.24/0.28	39.6/41.99	42.39/45.08
128	Kyber768 (ref) Saber (Ref)	0.14/0.17	1.13/1.16	0.11/0.12	10.11/10.46	11.49/11.91
64	Kyber768 (ref) Saber (Ref)	0.05/0.05	0.57/0.59	0.05/0.06	2.45/2.54	3.12/3.24
32	Kyber768 (ref) Saber (Ref)	0.02/0.02	0.29/0.3	0.04/0.04	0.61/0.63	0.96/0.99
16	Kyber768 (ref) Saber (Ref)	0.01/0.01	0.14/0.16	0.02/0.03	0.15/0.16	0.32/0.36
8	Kyber768 (ref) Saber (Ref)	0.0/0.0	0.07/0.08	0.02/0.02	0.04/0.04	0.13/0.14
4	Kyber768 (ref) Saber (Ref)	0.0/0.0	0.03/0.04	0.01/0.02	0.01/0.01	0.05/0.07
2	Kyber768 (ref) Saber (Ref)	0.0/0.0	0.02/0.02	0.01/0.02	0.0/0.0	0.03/0.04
2048	Kyber1024 (ref) FireSaber (Ref)	29.72/38.13	29.91 /30.42	34.22/20.38	3983.69/4202.08	4077.54/4291.01
1024	Kyber1024 (ref) FireSaber (Ref)	7.06 /5.23	16.34 /13.47	3.73/2.95	1183.47/829.28	1210.6/850.93
512	Kyber1024 (ref) FireSaber (Ref)	2.18 /1.56	8.18/6.01	1.22/1.0	293.93/213.52	305.51/222.09
256	Kyber1024 (ref) FireSaber (Ref)	0.7/0.5	4.14/3.12	0.43/0.34	74.84/53.75	80.11/57.71
128	Kyber1024 (ref) FireSaber (Ref)	0.25/0.2	2.14/1.65	0.19/0.15	18.54/13.39	21.12/15.39

64	Kyber1024 (ref) FireSaber (Ref)	0.1/0.07	1.07/0.76	0.1/0.07	4.64/3.45	5.91/4.35
32	Kyber1024 (ref) FireSaber (Ref)	0.04 /0.03	0.58/0.39	0.05/0.04	1.17/0.86	1.84/1.32
16	Kyber1024 (ref) FireSaber (Ref)	0.02 /0.01	0.3/0.2	0.03/0.03	0.3/0.22	0.65/0.46
8	Kyber1024 (ref) FireSaber (Ref)	0.01 /0.01	0.16/0.09	0.03/0.02	0.08/0.05	0.28/0.17
4	Kyber1024 (ref) FireSaber (Ref)	0.01 /0.0	0.1/0.05	0.02/0.02	0.02/0.01	0.15/0.08
2	Kyber1024 (ref) FireSaber (Ref)	0.0/0.0	0.04/0.02	0.02/0.01	0.01/0.0	0.07/0.03

Table 5. 8 Comparison of the time taken per round at various security levels based on the number of parties secured under the QROM

Parties	security	init_time (ref/Ref)	Round12 (ref/Ref)	Round3(ref/Ref)	Round4(ref/Ref)	total_time (ref/Ref)
2048	Kyber512 (ref) LightSaber (Ref)	13.46/26.64	9.11/9.4	6.85/8.12	1472.3/1257.24	1501.72/1301.4
1024	Kyber512 (ref) LightSaber (Ref)	3.17/3.44	4.53/3.32	1.54/1.49	357.2/271.79	366.44/280.04
512	Kyber512 (ref) LightSaber (Ref)	0.97/0.95	2.37/1.53	0.45/0.4	88.11/67.0	91.9/69.88
256	Kyber512 (ref) LightSaber (Ref)	0.29/0.31	1.12/0.79	0.16/0.14	21.74/17.47	23.31/18.71
128	Kyber512 (ref) LightSaber (Ref)	0.09/0.08	0.54/0.45	0.07/0.07	5.36/4.18	6.06/4.78
64	Kyber512 (ref)	0.03/0.03	0.28/0.22	0.04/0.04	1.35/1.08	1.7/1.37

	LightSaber (Ref)					
32	Kyber512 (ref)	0.01/0.01	0.15/0.1	0.03/0.02	0.35/0.27	0.54/0.4
	LightSaber (Ref)					
16	Kyber512 (ref)	0.01/0.0	0.08/0.05	0.02/0.02	0.09/0.07	0.2/0.14
	LightSaber (Ref)					
8	Kyber512 (ref)	0.0/0.0	0.03/0.02	0.02/0.01	0.02/0.02	0.07/0.05
	LightSaber (Ref)					
4	Kyber512 (ref)	0.0/0.0	0.02/0.01	0.01/0.01	0.01/0.0	0.04/0.02
	LightSaber (Ref)					
2	Kyber512 (ref)	0.0/0.0	0.01/0.01	0.01/0.01	0.0/0.0	0.02/0.02
	LightSaber (Ref)					
2048	Kyber768 (ref)	19.55/27.53	17.58/13.25	9.53/8.93	2782.74/1976.13	2829.4/2025.84
	Saber (Ref)					
1024	Kyber768 (ref)	5.31/4.06	8.77/5.6	2.56/2.15	701.74/486.38	718.38/498.19
	Saber (Ref)					
512	Kyber768 (ref)	1.48/1.44	4.1/3.35	0.73/0.7	165.16/139.24	171.47/144.73
	Saber (Ref)					
256	Kyber768 (ref)	0.42/0.46	1.93/1.67	0.24/0.26	40.84/35.07	43.43/37.46
	Saber (Ref)					
128	Kyber768 (ref)	0.13/0.16	0.92/0.86	0.11/0.11	8.67/8.71	9.83/9.84
	Saber (Ref)					
64	Kyber768 (ref)	0.04/0.05	0.44/0.43	0.05/0.05	2.11/2.17	2.64/2.7
	Saber (Ref)					
32	Kyber768 (ref)	0.02/0.02	0.22/0.23	0.03/0.03	0.53/0.54	0.8/0.82
	Saber (Ref)					
16	Kyber768 (ref)	0.01/0.01	0.11/0.12	0.02/0.03	0.13/0.13	0.27/0.29
	Saber (Ref)					
8	Kyber768 (ref)	0.0/0.0	0.05/0.05	0.02/0.02	0.04/0.04	0.11/0.11
	Saber (Ref)					
4	Kyber768 (ref)	0.0/0.0	0.03/0.03	0.01/0.02	0.01/0.01	0.05/0.06

	Saber (Ref)					
2	Kyber768 (ref) Saber (Ref)	0.0/0.0	0.01/0.02	0.01/0.02	0.0/0.0	0.02/0.04
2048	Kyber1024 (ref) FireSaber (Ref)	29.33/27.9	25.47/22.04	38.55/17.43	4001.39/3234.33	4094.74/3301.7
1024	Kyber1024 (ref) FireSaber (Ref)	6.8/4.97	12.85/9.29	3.58/3.28	1045.61/752.9	1068.84/770.44
512	Kyber1024 (ref) FireSaber (Ref)	2.05/1.64	6.66/4.53	1.13/0.94	262.1/190.1	271.94/197.21
256	Kyber1024 (ref) FireSaber (Ref)	0.67/0.51	3.36/2.41	0.4/0.31	65.66/47.7	70.09/50.93
128	Kyber1024 (ref) FireSaber (Ref)	0.27/0.2	1.66/1.23	0.18/0.15	16.33/12.02	18.44/13.6
64	Kyber1024 (ref) FireSaber (Ref)	0.12/0.06	0.87/0.61	0.08/0.08	4.09/3.09	5.16/3.84
32	Kyber1024 (ref) FireSaber (Ref)	0.04/0.03	0.49/0.29	0.05/0.04	1.06/0.75	1.64/1.11
16	Kyber1024 (ref) FireSaber (Ref)	0.02/0.01	0.24/0.17	0.04/0.03	0.27/0.22	0.57/0.43
8	Kyber1024 (ref) FireSaber (Ref)	0.01/0.01	0.14/0.07	0.03/0.02	0.08/0.05	0.26/0.15
4	Kyber1024 (ref) FireSaber (Ref)	0.0/0.0	0.07/0.04	0.03/0.02	0.02/0.01	0.12/0.07
2	Kyber1024 (ref) FireSaber (Ref)	0.0/0.0	0.03/0.02	0.02/0.01	0.0/0.0	0.05/0.03

5.2 Utilizing Saber'.2AKE on ARM Group Key Exchange Protocol:

Developing post-quantum cryptographic algorithms has opened opportunities for secure communication in environments with limited resources, such as the IoT. A promising algorithm called Saber, a lattice-based KEM, has shown practicality for IoT

devices when implemented on ARM processors [102]. This section demonstrates the theoretical foundations of incorporating Saber.KEM results on the ARM processor are integrated into a GKE protocol and offer a comprehensive performance analysis.

The clock cycle counts and memory consumption values for the Saber KEM implementation on ARM Cortex-M processors are as follows:

$$Time\ (t) = {clock\ cycle(C)}/{clock\ speed\ (f)}$$

Table 5. 9 Cryptographic operations on ARM processor

	Cortex-M4 ($f = 168\ MHz$)			**Cortex-M0** ($f = 32\ MHz$)		
	Clock cycle (C)	**Time (t)**	**Memory**	**Clock cycle (C)**	**Time (t)**	**Memory**
Key Generation	1,147 x 10^3	$\approx 0.007\ ms$	≈ 6.76 KB	4,786 x 10^3	$\approx 0.15\ ms$	≈ 4.89 KB
Encapsulation	1,444 x 10^3	$\approx 0.008\ ms$	≈ 5.84 KB	6,328 x 10^3	$\approx 0.20\ ms$	≈ 3.99 KB
Decapsulation	1,543 x 10^3	$\approx 0.009\ ms$	≈ 6.99 KB	7,509 x 10^3	$\approx 0.23\ ms$	≈ 5.05 KB

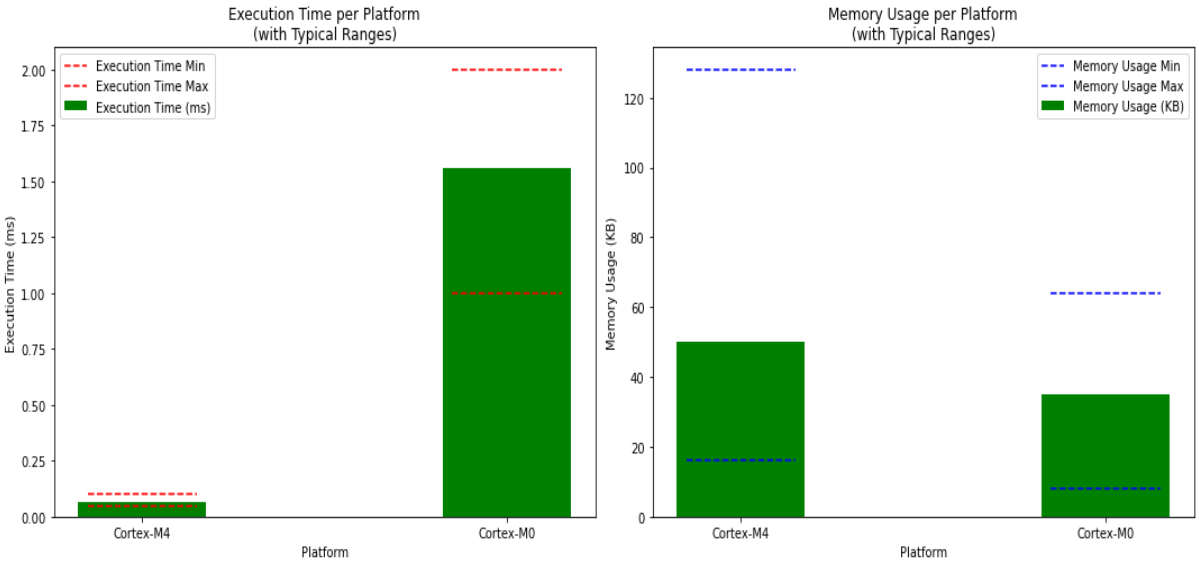

Figure 5. 6 Performance Analysis of Execution Time and Memory Usage on Cortex-M0 and Cortex-M4 Processors

5.2.1 Saber'.2AKE and its Suitability for IoTs:

Recent implementations of Saber on ARM Cortex-M4 processors have demonstrated the algorithm's practicality in IoT environments, with low latency and minimal memory consumption[102]. Our proposed GKE protocol utilizes Saber'.2AKE for secure pairwise key exchanges between neighboring nodes, combined to derive the group key.

1. Mapping Performance on GKE Protocol:

Rounds 1-2: Each participant U_i executes Saber'.2AKE with its neighbors U_{i-1} and U_{i+1} , resulting in pairwise shared keys $\vec{K_i}$ and $\overleftarrow{K_i}$

- **Key Generation**: t_{keygen}, **Encapsulation**: $2 \times t_{Encaps}$ &
- **Decapsulation**: $2 \times t_{Decaps}$

  Round 3: Requires the generation of a commitment using Saber.PKE, a form of public-key encryption. Assume its performance metrics are equivalent to Saber'.2AKE encapsulation.

- **Encapsulation**: $2 \times t_{Encaps}$

Round 4: Broadcasts and verification are lightweight as compared to cryptographic operations.

2. Computational Efficiency:

The performance of the Saber'.2AKE operations significantly impact the computational efficiency of the Saber-GKE protocol. Based on the results obtained from implementing the code on ARM Cortex-M4 processors (as shown in Table. 5.9), the times for encapsulation and decapsulation are as follows:

For each participant, the T_{total} is:

$$T_{total} = (t_{keygen} + 4 \times t_{Encaps} + 2 \times t_{Decaps})$$

- **ARM Cortex-M4 Platform**:

$$T_{total} = (0.007\ ms + 4 \times 0.008\ ms + 2 \times 0.009\ ms)$$

$$\approx 0.057\ ms$$

- **ARM Cortex-M0 Platform**:

$$T_{total} = (0.15\ ms + 4 \times 0.20\ ms + 2 \times 0.23\ ms)$$

$$\approx 1.41\ ms$$

3. Memory Usage:

The memory usage is another critical factor for IoT devices. The memory requirements for the Saber-GKE operations are:

Thus, the total memory usage for each participant is:

$$M_{total} = (m_{keygen} + 4\ \times m_{Encap} + 2 \times m_{Decap})$$

- **ARM Cortex-M4 Platform**:

$$M_{total} = (6.76\ KB + 4 \times 5.84\ KB + 2 \times 6.99\ KB)$$

$$\approx 44.1\ KB$$

- **ARM Cortex-M0 Platform**:

$$M_{total} = (4.89\ KB + 4 \times 3.99\ KB + 2 \times 5.05\ KB)$$

$$\approx 30.95\ KB$$

5.2.2 Application of Proposed Scheme in IoTs:

The Saber-GAKE Protocol is ideal for secure group communication in IoT settings. It addresses the issues quantum computing poses while facilitating efficient and scalable key exchanges [22], [103]. In smart grid systems, like Advanced Metering Infrastructure (AMI) and Supervisory Control and Data Acquisition (SCADA) systems, the protocol facilitates secure data exchange among smart meters, sensors, and controllers with minimal computational overhead, essential for the integrity and efficiency of critical infrastructure[11].

Additionally, in healthcare applications, particularly inside Wireless Body Area Networks (WBANs), the protocol facilitates the safe communication of sensitive patient data from wearable devices and sensors to healthcare practitioners. This guarantees anonymity and security in resource-limited contexts, where minimal power consumption and efficient communication protocols are crucial. Furthermore, in Intelligent Transportation Systems (ITS), the protocol enables secure group communication among cars and infrastructure (e.g., Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications), ensuring strong protection in dynamic settings [104].

The Saber-GAKE protocol is efficient and practical for IoT devices, with execution times and memory consumption significantly within the permitted limits (Figure 5.6). In conjunction with its robust quantum resistance, these performance metrics demonstrate the protocol's suitability for future post-quantum cryptographic applications in IoT.

5.2.3 Limitation of Performance Estimation:

It is important to note that the execution time and memory consumption provided for ARM Cortex-M4 and M0 platforms in section 5.2.1 are based on analytical estimations derived from prior Saber implementation benchmarks, rather than direct implementation of the full Saber-GAKE protocol on actual hardware. While these estimations offer meaningful insights into the feasibility of deployment in constrained IoT environments, they do not account for miscellaneous protocol-specific operations such as control flow logic, session management overhead, buffer allocations, and communication delays, which could add non-trivial computational and memory costs.

Furthermore, our estimations assume that all cryptographic operations (e.g., key generation, encapsulation, decapsulation) execute in isolated conditions with no external system-level interference. In practice, IoT devices often run multiple concurrent tasks, which can impact both timing and memory availability.

As such, while the presented protocol is strongly suggested for suitability for IoT deployment, a full implementation on physical hardware is recommended for precise performance profiling under real-world conditions.

5.3 Summary

In this chapter, we investigated the Saber-GAKE protocol's performance compared to the Compiled Kyber implementation. The chapter describes the experimental setup and methodology used to analyse both protocols in ROM and QROM. This comparison shows that Saber-GAKE is more efficient in cryptographic activities such as key generation, encapsulation, and decapsulation while maintaining a high level of quantum resistance.

We also described the practical application of Saber'.2AKE on resource-constrained devices, notably ARM Cortex-M CPUs. The protocol performed exceptionally well, with execution times as low as 0.064 ms on ARM Cortex-M4 and 1.56 ms on Cortex-M0. Furthermore, its memory utilisation stayed well under hardware restrictions, demonstrating its appropriateness for IoT contexts with constrained computing resources. It emphasises Saber-GAKE's capacity to efficiently manage secure communication for up to 2000 people, demonstrating its scalability and applicability for a wide range of IoT applications, including smart grids and intelligent transportation systems. This chapter proves Saber-GAKE as a reliable and realistic post-quantum security solution for IoT networks by assessing its computational efficiency, scalability, and minimal resource requirements.

This chapter thoroughly explains the protocol's capabilities and practical applications, laying the groundwork for its implementation in various real-world circumstances where secure group communication is critical.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

This research has effectively tackled the critical need for group key management methods capable of resisting the imminent threats from quantum computing, especially in the context of Internet of Things (IoT) environments. The rapid progress in quantum technology underscores the necessity of creating cryptographic solutions that can protect data from quantum threats. Considering these problems, our endeavour commences with an exhaustive examination of current group key management practices. This thorough analysis exposes considerable deficiencies in their ability to guarantee post-quantum security, highlighting the need for novel strategies. In response to this requirement, we devised the Saber-GAKE protocol, an innovative group key management solution based on lattice-based cryptography. This protocol's design has been rigorously assessed to ensure it surpasses the demanding security standards of the post-quantum era. Our comprehensive investigation has shown that the Saber-GAKE protocol performs exceptionally well in situations requiring substantial group communication, effectively supporting up to 2000 people. When directly compared to Pablos' "Compiled Kyber," the Saber-GAKE protocol demonstrates enhanced computing performance and resource efficiency, distinguishing itself within its category.

The protocol demonstrates exceptional execution durations of about 0.064 milliseconds on ARM Cortex-M4 and 1.56 milliseconds on ARM Cortex-M0 systems, while utilising just 50.1 KB and 35.85 KB of memory, respectively. These performance measurements highlight its suitability for the resource-limited characteristics of standard IoT devices, guaranteeing strong post-quantum security

without excessively taxing system resources. This efficiency is essential for IoT applications, as devices frequently function under severe constraints regarding processing power and memory.

The Saber-GAKE protocol is founded on the Module-Learning With Rounding (Module-LWR) assumption in the Quantum Random Oracle Model (QROM), offering robust theoretical security assurances against conventional and quantum adversaries. This foundation guarantees that the protocol remains robust under the most rigorous security assessments. The Saber-GAKE protocol utilises lattice-based cryptographic concepts to achieve a commendable equilibrium between robust security and maximum efficiency, rendering it especially appropriate for the distinct needs of IoT networks.

Furthermore, the adaptability of the Saber-GAKE protocol enables its implementation in a diverse range of IoT applications, hence augmenting its practical utility. In smart grid systems, it enables secure communication between smart meters and controllers, thereby ensuring the integrity and efficiency of essential infrastructure. In healthcare, specifically in Wireless Body Area Networks (WBANs), the protocol safeguards sensitive patient information transmitted between wearable devices and healthcare providers, while simultaneously reducing power consumption to extend device lifespan. Moreover, in intelligent transportation systems (ITS), the protocol guarantees safe group communication between cars and infrastructure, accommodating dynamic, high-velocity situations with rigorous security requirements.

The Saber-GAKE protocol's resilient architecture and thorough examination offer a pragmatic and efficient approach for administering post-quantum group keys in IoT networks. The protocol has undergone rigorous testing and assessment, demonstrating its capacity to endure quantum computing threats while ensuring practicality for real-

world implementation. The lightweight architecture facilitates seamless operation in IoT devices with constrained computational and memory resources, providing robust protection against quantum attacks while maintaining performance integrity. It is crucial to recognise that post-quantum cryptography is an emerging domain. With the advancement of quantum technology and the emergence of new threats, ongoing research and development are essential to adapt to the evolving security landscape. Subsequent research endeavours will concentrate on implementing the Saber-GAKE protocol on real IoT platforms to enhance its architecture. This will encompass modifications to facilitate centralised, decentralised, and distributed systems, thereby augmenting the protocol's robustness and relevance across various real-world contexts. In the end, the Saber-GAKE protocol signifies a substantial advancement in post-quantum cryptography solutions for IoT applications. Its proven computational efficiency, strong security assurances, and compatibility with resource-constrained platforms make it a vital advancement for secure group communication in the post-quantum age. This study establishes a robust basis for future progress, guaranteeing the preservation of secure group communication amid the evolving quantum threat landscape.

6.2 Future Work

The Saber-GAKE protocol shows potential in its design, analysis, and efficiency, especially given the limitations of existing IoT frameworks. Nevertheless, numerous areas necessitate additional research and development to realize its promise in practical applications completely. These domains encompass architectural advances, performance improvements, and sophisticated security features, all essential for adapting the protocol to the changing landscape of IoT and quantum-resistant encryption.

6.2.1 Architectural Improvements

A primary future goal entails investigating and enhancing the Saber-GAKE protocol across several architectural configurations—centralized, decentralised, and distributed. In centralised infrastructures, where a singular trusted authority oversees group communication, the emphasis will be on enhancing efficiency, especially in low-latency contexts. Centralised setups may benefit from enhanced group management solutions that optimise operations like key distribution and member authentication, hence minimising communication overhead and delay.

In contrast, decentralised and distributed architectures provide distinct issues. In decentralised systems, devoid of a singular control point, the protocol must be modified to address challenges such as scalability, necessitating efficient support for an expanding number of nodes without performance deterioration. Furthermore, resilience to node failures is essential, necessitating the protocol to uphold secure communication despite the unresponsiveness or compromise of specific nodes. Another element is dynamic group membership, characterised by nodes frequently joining and departing from the network. The protocol must effectively manage these modifications without jeopardising continuing communication or substantially elevating computing demands. These enhancements are crucial for applications in highly dynamic settings such as automotive networks, where nodes are perpetually in motion, or peer-to-peer IoT ecosystems, characterised by ad hoc and spontaneous device interactions.

6.2.2 Enhancement of Performance

A vital focus for future endeavours is the enhancement of the protocol's performance metrics. The existing approach exhibits remarkable execution times and memory efficiency, especially on ARM Cortex systems, frequently utilised in IoT devices.

Nonetheless, there exists potential for additional enhancement. Hardware-specific optimisations may be investigated to utilise the distinct capabilities of various IoT hardware platforms. Custom optimisations designed for certain processors may produce substantial performance improvements. Moreover, hybrid approaches that integrate software and hardware acceleration, particularly those employing quantum-safe hardware elements, may significantly improve efficiency. These solutions may alleviate the protocol's computing demands, enhancing its applicability for resource-limited devices such as smart sensors or wearable technologies.

6.2.3 Enhanced Security Capabilities

The changing threat landscape requires incorporating sophisticated security elements into the Saber-GAKE protocol. Subsequent research must focus on forward secrecy, particularly in dynamic group settings characterised by frequent alterations in group membership. Forward secrecy guarantees that prior communications remain safe even if long-term keys are compromised. This functionality is essential for preserving the integrity of sensitive data over time, especially in IoT applications related to critical infrastructure or personal information.

Resistance to side-channel attacks is another critical domain. IoT devices, frequently situated in uncontrolled areas, are vulnerable to physical assaults that take advantage of information loss, such power usage or electromagnetic emissions. Formulating defences against such assaults, especially in hardware implementations, will be essential for guaranteeing the protocol's resilience.

Ultimately, using advanced authentication procedures could substantially improve the protocol's security. These approaches may encompass multi-factor authentication or biometric systems, enhancing security beyond conventional cryptographic methods.

Achieving seamless integration of these functionalities without sacrificing performance or usability would be a significant task.

6.2.4 Pragmatic Implementation and Flexibility

Its practical implementation in real-world contexts must be rigorously examined to ascertain the Saber-GAKE protocol's applicability in forthcoming IoT applications. This entails modifying the protocol to accommodate emerging cryptographic innovations, including new quantum-resistant algorithms, and guaranteeing its alignment with forthcoming IoT standards and protocols. Furthermore, facilitating compatibility with diverse IoT devices, including low-power sensors and high-performance edge devices, will be crucial for extensive adoption.

Incorporating these innovations will guarantee that the Saber-GAKE protocol stays in the vanguard of secure group communication, offering a resilient, scalable, and efficient solution for the evolving domains of IoT and post-quantum cryptography.

REFERENCES

[1] M. Dammak, S. M. Senouci, M. A. Messous, M. H. Elhdhili, and C. Gransart, "Decentralized Lightweight Group Key Management for Dynamic Access Control in IoT Environments," *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1742–1757, Sep. 2020, doi: 10.1109/TNSM.2020.3002957.

[2] G. Kaur and K. S. Saini, "Securing Network Communication Between Motes Using Hierarchical Group Key Management Scheme Using Threshold Cryptography in Smart Home Using Internet of Things," *Lecture Notes in Networks and Systems*, vol. 12, pp. 201–212, 2017, doi: 10.1007/978-981-10-3935-5_21.

[3] "41.6 billion IoT devices will be generating 79.4 zettabytes of data in 2025 - Help Net Security." Accessed: Feb. 29, 2024. [Online]. Available: https://www.helpnetsecurity.com/2019/06/21/connected-iot-devices-forecast/

[4] A. Albakri and L. Harn, "Non-Interactive Group Key Pre-Distribution Scheme (GKPS) for End-To-End Routing in Wireless Sensor Networks," *IEEE Access*, vol. 7, pp. 31615–31623, 2019, doi: 10.1109/ACCESS.2019.2900390.

[5] "How 5G, AI And IoT Are Set To Accelerate Digital Transformation." Accessed: Jan. 05, 2025. [Online]. Available:

https://www.forbes.com/councils/forbeslacouncil/2019/05/23/how-5g-ai-and-iot-are-set-to-accelerate-digital-transformation/

[6] G. Perrone, M. Vecchio, R. Pecori, and R. Giaffreda, "The Day After Mirai: A Survey on MQTT Security Solutions After the Largest Cyber-attack Carried Out through an Army of IoT Devices," *International Conference on Internet of Things, Big Data and Security*, pp. 246–253, 2017, doi: 10.5220/0006287302460253.

[7] Md. T. Islam and B. U. Khan, "Big Data and Analytics: Prospects, Challenges, and the Way Forward," *https://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-6684-7366-5.ch048*, pp. 1–30, Jan. 1AD, doi: 10.4018/978-1-6684-7366-5.CH048.

[8] I. Hedi, I. Špeh, and A. Šarabok, "IoT network protocols comparison for the purpose of IoT constrained networks," *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2017 - Proceedings*, pp. 501–505, Jul. 2017, doi: 10.23919/MIPRO.2017.7973477.

[9] S. Nehal and M. Farhan, "QUANTUM CRYPTOGRAPHY - BREAKING RSA ENCRYPTION USING QUANTUM COMPUTING WITH SHOR'S ALGORITHM," 2020.

[10] K. Csenkey and N. Bindel, "Post-quantum cryptographic assemblages and the governance of the quantum threat," *J Cybersecur*, vol. 2023, pp. 1–14, doi: 10.1093/cybsec/tyad001.

[11] F. Samiullah, M. L. Gan, S. Akleylek, and Y. Aun, "Group Key Management in Internet of Things: A Systematic Literature Review," *IEEE Access*, vol. 11, pp. 77464–77491, 2023, doi: 10.1109/ACCESS.2023.3298024.

[12] Q. Cheng, C. Hsu, and L. Harn, "Lightweight Noninteractive Membership Authentication and Group Key Establishment for WSNs," *Math Probl Eng*, vol. 2020, no. 1, p. 1452546, Jan. 2020, doi: 10.1155/2020/1452546.

[13] Y. Hanna, M. Cebe, S. Mercan, and K. Akkaya, "Efficient Group-Key Management for Low-bandwidth Smart Grid Networks," *2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids, SmartGridComm 2021*, pp. 188–193, 2021, doi: 10.1109/SMARTGRIDCOMM51999.2021.9631988.

[14] P. Porambage, A. Braeken, C. Schmitt, A. Gurtov, M. Ylianttila, and B. Stiller, "Group key establishment for enabling secure multicast communication in wireless sensor networks deployed for IoT applications," *IEEE Access*, vol. 3, pp. 1503–1511, 2015, doi: 10.1109/ACCESS.2015.2474705.

[15] O. Cheikhrouhou, "Secure Group Communication in Wireless Sensor Networks: A survey," *Journal of Network and Computer Applications*, vol. 61, pp. 115–132, Feb. 2016, doi: 10.1016/J.JNCA.2015.10.011.

[16] D. K. Ajmani, A. Srivastava, J. D'Souza, and J. Abraham, "Homomorphic Multicast Group Key Management: Using Routing Protocol for Low Power and Lossy Networks,"

Proceedings of the International Conference on Intelligent Computing and Control Systems, ICICCS 2020, pp. 6–11, May 2020, doi: 10.1109/ICICCS48265.2020.9120954.

[17] A. Piccoli, M. O. Pahl, and L. Wüstrich, "Group Key Management in constrained IoT Settings," *Proc IEEE Symp Comput Commun*, vol. 2020-July, Jul. 2020, doi: 10.1109/ISCC50000.2020.9219619.

[18] M. A. Kandi, H. Lakhlef, A. Bouabdallah, and Y. Challal, "An Efficient Multi-Group Key Management Protocol for Heterogeneous IoT Devices," *IEEE Wireless Communications and Networking Conference, WCNC*, vol. 2019-April, Apr. 2019, doi: 10.1109/WCNC.2019.8885613.

[19] N. G. Felde, T. Guggemos, T. Heider, and D. Kranzlmuller, "Secure group key distribution in constrained environments with IKEv2," *2017 IEEE Conference on Dependable and Secure Computing*, pp. 384–391, Oct. 2017, doi: 10.1109/DESEC.2017.8073823.

[20] A. Kabra, S. Kumar, and G. S. Kasbekar, "Efficient, Flexible and Secure Group Key Management Protocol for Dynamic IoT Settings," *EAI Endorsed Transactions on Internet of Things*, vol. 7, no. 25, p. 168862, Aug. 2020, doi: 10.4108/eai.3-3-2021.168862.

[21] H. Gu and M. Potkonjak, "Efficient and secure group key management in IoT using multistage interconnected PUF," *Proceedings of the International Symposium on Low Power Electronics and Design*, Jul. 2018, doi: 10.1145/3218603.3218646.

[22] T. Liu, G. Ramachandran, and R. Jurdak, "Post-Quantum Cryptography for Internet of Things: A Survey on Performance and Optimization," Jan. 2024, Accessed: Oct. 11, 2024. [Online]. Available: https://arxiv.org/abs/2401.17538v1

[23] F. Samiullah, M. L. Gan, S. Akleylek, and Y. Aun, "Post-Quantum Group Key Management in IoTs," *2023 25th International Multi Topic Conference, INMIC 2023 - Proceedings*, 2023, doi: 10.1109/INMIC60434.2023.10466001.

[24] R. Asif, "Post-Quantum Cryptosystems for Internet-of-Things: A Survey on Lattice-Based Algorithms," *IoT 2021, Vol. 2, Pages 71-91*, vol. 2, no. 1, pp. 71–91, Feb. 2021, doi: 10.3390/IOT2010005.

[25] A. Lohachab, A. Lohachab, and A. Jangra, "A comprehensive survey of prominent cryptographic aspects for securing communication in post-quantum IoT networks," *Internet of Things*, vol. 9, p. 100174, Mar. 2020, doi: 10.1016/J.IOT.2020.100174.

[26] T. Liu, G. Ramachandran, and R. Jurdak, "Post-Quantum Cryptography for Internet of Things: A Survey on Performance and Optimization," Jan. 2024, Accessed: Apr. 29, 2024. [Online]. Available: https://arxiv.org/abs/2401.17538v1

[27] "Cryptography 101: From Theory to Practice - Rolf Oppliger - Google Books." Accessed: Jan. 08, 2025. [Online]. Available: https://books.google.com.my/books?hl=en&lr=&id=ET86EAAAQBAJ&oi=fnd&pg=PR7&dq=Cryptography+Basics&ots=2wm9EirGrI&sig=Ysj3QYE4XSVoIOUciOBH_GVe4rk&redir_esc=y#v=onepage&q=Cryptography%20Basics&f=false

[28] S. Kumar, M. S. Gaur, P. Sagar Sharma, and D. Munjal, “A Novel Approach of Symmetric Key Cryptography,” *Proceedings of 2021 2nd International Conference on Intelligent Engineering and Management, ICIEM 2021*, pp. 593–598, Apr. 2021, doi: 10.1109/ICIEM51511.2021.9445343.

[29] M. S. A. Mohamad, R. Din, and J. I. Ahmad, “Research trends review on RSA scheme of asymmetric cryptography techniques,” *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 1, pp. 487–492, Feb. 2021, doi: 10.11591/EEI.V10I1.2493.

[30] T. Gebremichael, U. Jennehag, and M. Gidlund, “Lightweight IoT Group Key Establishment Scheme Using One-way Accumulator,” *2018 International Symposium on Networks, Computers and Communications, ISNCC 2018*, Nov. 2018, doi: 10.1109/ISNCC.2018.8531034.

[31] Y. H. Kung and H. C. Hsiao, “GroupIt: Lightweight Group Key Management for Dynamic IoT Environments,” *IEEE Internet Things J*, vol. 5, no. 6, pp. 5155–5165, Dec. 2018, doi: 10.1109/JIOT.2018.2840321.

[32] T. Prantl *et al.*, “Towards a Group Encryption Scheme Benchmark: A View on Centralized Schemes with Focus on IoT,” *ICPE 2021 - Proceedings of the ACM/SPEC International Conference on Performance Engineering*, pp. 233–240, Apr. 2021, doi: 10.1145/3427921.3450252.

[33] T. M. Fernandez-Carames, “From Pre-Quantum to Post-Quantum IoT Security: A Survey on Quantum-Resistant Cryptosystems for the Internet of Things,” *IEEE Internet Things J*, vol. 7, no. 7, pp. 6457–6480, Jul. 2020, doi: 10.1109/JIOT.2019.2958788.

[34] S. Iqbal, M. L. Mat Kiah, A. ur Rehman, Z. Abbas, and B. Daghighi, “DM-GKM: A key management scheme for dynamic group based applications,” *Computer Networks*, vol. 182, p. 107476, Dec. 2020, doi: 10.1016/J.COMNET.2020.107476.

[35] E. Abirami and T. Padmavathy, “Proficient key management scheme for multicast groups using group key agreement and broadcast encryption,” *2017 International Conference on Information Communication and Embedded Systems, ICICES 2017*, Oct. 2017, doi: 10.1109/ICICES.2017.8070789.

[36] C. L. Hsu and T. V. Le, “A Time Bound Dynamic Group key Distribution Scheme with Anonymous Three-factor Identification for IoT-Based Multi-Server Environments,” *Proceedings - 2020 15th Asia Joint Conference on Information Security, AsiaJCIS 2020*, pp. 59–65, Aug. 2020, doi: 10.1109/ASIAJCIS50894.2020.00021.

[37] Q. Cheng, C. Hsu, Z. Xia, and L. Harn, “Fast multivariate-polynomial-based membership authentication and key establishment for secure group communications in WSN,” *IEEE Access*, vol. 8, pp. 71833–71839, 2020, doi: 10.1109/ACCESS.2020.2987978.

[38] G. Alagic *et al.*, “Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process,” Jan. 2019, doi: 10.6028/NIST.IR.8240.

[39] V. Vedral and M. B. Plenio, “Basics of quantum computation,” *Prog Quantum Electron*, vol. 22, no. 1, pp. 1–39, Jan. 1998, doi: 10.1016/S0079-6727(98)00004-4.

[40] E. Rieffel and W. Polak, “An introduction to quantum computing for non-physicists,” *ACM Computing Surveys (CSUR)*, vol. 32, no. 3, pp. 300–335, Sep. 2000, doi: 10.1145/367701.367709.

[41] “SABER: LWR-based KEM.” Accessed: Oct. 30, 2023. [Online]. Available: https://www.esat.kuleuven.be/cosic/pqcrypto/saber/

[42] R. Avanzi *et al.*, “CRYSTALS-Kyber Algorithm Specifications And Supporting Documentation (version 3.02),” 2021.

[43] S. Bai *et al.*, “Crystals-Dilithium Round 3 Update,” 2021.

[44] N. Gupta, A. Jati, A. Chattopadhyay, and G. Jha, “Lightweight Hardware Accelerator for Post-Quantum Digital Signature CRYSTALS-Dilithium,” *Cryptology ePrint Archive*, 2022, Accessed: Dec. 17, 2024. [Online]. Available: https://eprint.iacr.org/2022/496

[45] G. Alagic *et al.*, “Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process,” Jul. 2022, doi: 10.6028/NIST.IR.8413.

[46] J. Ding, S. Alsayigh, J. Lancrenon, S. Rv, and M. Snook, “Provably Secure Password Authenticated Key Exchange Based on RLWE for the Post-Quantum World”.

[47] R. Barskar and M. Chawla, “A Survey on Efficient Group Key Management Schemes in Wireless Networks,” *Indian J Sci Technol*, vol. 9, no. 14, pp. 1–16, Apr. 2016, doi: 10.17485/IJST/2016/V9I14/87972.

[48] A. Ghafoor, M. Sher, M. Imran, and K. Saleem, “A Lightweight Key Freshness Scheme for Wireless Sensor Networks,” *Proceedings - 12th International Conference on Information Technology: New Generations, ITNG 2015*, pp. 169–173, May 2015, doi: 10.1109/ITNG.2015.32.

[49] “IEEE Xplore Full-Text PDF:” Accessed: Dec. 17, 2024. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7053290

[50] P. Szalachowski and T. H. J. Kim, “Secure broadcast in distributed networks with strong adversaries,” *Security and Communication Networks*, vol. 8, no. 18, pp. 3739–3750, Dec. 2015, doi: 10.1002/SEC.1296.

[51] P. Sharma and P. B R, “A lightweight group key management scheme with constant rekeying cost and public bulletin size,” *Information Security Journal: A Global Perspective*, vol. 33, no. 2, pp. 97–120, Mar. 2024, doi: 10.1080/19393555.2023.2198737.

[52] “A Decentralized Batch-Based Group Key Management Protocol for Mobile Internet of Things (DBGK) | IEEE Conference Publication | IEEE Xplore.” Accessed: Dec. 17, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/7363210

[53] H. Tan and I. Chung, “Secure Authentication and Group Key Distribution Scheme for WBANs Based on Smartphone ECG Sensor,” *IEEE Access*, vol. 7, pp. 151459–151474, 2019, doi: 10.1109/ACCESS.2019.2948207.

[54] M. A. Mughal, P. Shi, A. Ullah, K. Mahmood, M. Abid, and X. Luo, "Logical Tree Based Secure Rekeying Management for Smart Devices Groups in IoT Enabled WSN," *IEEE Access*, vol. 7, pp. 76699–76711, 2019, doi: 10.1109/ACCESS.2019.2921999.

[55] "A group key agreement protocol for intelligent internet of things system - Zhang - 2022 - International Journal of Intelligent Systems - Wiley Online Library." Accessed: Dec. 24, 2024. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/int.22644

[56] M. T. Dong and H. Xu, "Group Key Management Scheme for Multicast Communication Fog Computing Networks," *Processes 2020, Vol. 8, Page 1300*, vol. 8, no. 10, p. 1300, Oct. 2020, doi: 10.3390/PR8101300.

[57] U. Mustafa and N. Philip, "Group-Based Key Exchange for Medical IoT Device-to-Device Communication (D2D) Combining Secret Sharing and Physical Layer Key Exchange," *Proceedings of 12th International Conference on Global Security, Safety and Sustainability, ICGS3 2019*, Apr. 2019, doi: 10.1109/ICGS3.2019.8688022.

[58] H. Harb, A. William, O. A. El-Mohsen, and H. A. Mansour, "Multicast security model for Internet of Things based on context awareness," *ICENCO 2017 - 13th International Computer Engineering Conference: Boundless Smart Societies*, vol. 2018-January, pp. 303–309, Jul. 2017, doi: 10.1109/ICENCO.2017.8289805.

[59] S. Naskar, T. Zhang, G. Hancke, and M. Gidlund, "OTP-Based Symmetric Group Key Establishment Scheme for IoT Networks," *IECON Proceedings (Industrial Electronics Conference)*, vol. 2021-October, Oct. 2021, doi: 10.1109/IECON48115.2021.9590001.

[60] W. Song, M. Liu, T. Baker, Q. Zhang, and Y. an Tan, "A group key exchange and secure data sharing based on privacy protection for federated learning in edge-cloud collaborative computing environment," *International Journal of Network Management*, vol. 33, no. 5, p. e2225, Sep. 2023, doi: 10.1002/NEM.2225.

[61] A. Musuroi, B. Groza, L. Popa, and P. S. Murvay, "Fast and Efficient Group Key Exchange in Controller Area Networks (CAN)," *IEEE Trans Veh Technol*, vol. 70, no. 9, pp. 9385–9399, Sep. 2021, doi: 10.1109/TVT.2021.3098546.

[62] A. Lei, H. Cruickshank, Y. Cao, P. Asuquo, C. P. A. Ogah, and Z. Sun, "Blockchain-Based Dynamic Key Management for Heterogeneous Intelligent Transportation Systems," *IEEE Internet Things J*, vol. 4, no. 6, pp. 1832–1843, Dec. 2017, doi: 10.1109/JIOT.2017.2740569.

[63] A. Ghosal and M. Conti, "Key Management Systems for Smart Grid Advanced Metering Infrastructure: A Survey," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 3, pp. 2831–2848, Jul. 2019, doi: 10.1109/COMST.2019.2907650.

[64] V. Patil, V. Kulkarni, and H. Patil, "Improvised Group Key Management Protocol for SCADA System," *2018 International Conference on Smart City and Emerging Technology, ICSCET 2018*, Nov. 2018, doi: 10.1109/ICSCET.2018.8537287.

[65] O. B. J. Rabie, P. K. Balachandran, M. Khojah, and S. Selvarajan, "A Proficient ZESO-DRKFC Model for Smart Grid SCADA Security," *Electronics 2022, Vol. 11, Page 4144*, vol. 11, no. 24, p. 4144, Dec. 2022, doi: 10.3390/ELECTRONICS11244144.

[66] Z. Wang, R. Huo, and S. Wang, “A Lightweight Certificateless Group Key Agreement Method without Pairing Based on Blockchain for Smart Grid,” *Future Internet 2022, Vol. 14, Page 119*, vol. 14, no. 4, p. 119, Apr. 2022, doi: 10.3390/FI14040119.

[67] H. Nicanfar and V. C. M. Leung, “Password-authenticated cluster-based group key agreement for smart grid communication,” *Security and Communication Networks*, vol. 7, no. 1, pp. 221–233, Jan. 2014, doi: 10.1002/SEC.726.

[68] M. Benmalek and Y. Challal, “ESKAMI: Efficient and scalable multi-group key management for advanced metering infrastructure in smart grid,” *Proceedings - 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2015*, vol. 1, pp. 782–789, Dec. 2015, doi: 10.1109/TRUSTCOM.2015.447.

[69] M. Schnell, U. Epple, D. Shutin, and N. Schneckenburger, “LDACS: Future aeronautical communications for air-traffic management,” *IEEE Communications Magazine*, vol. 52, no. 5, pp. 104–110, 2014, doi: 10.1109/MCOM.2014.6815900.

[70] N. Maurer, T. Graupl, C. Schmitt, G. D. Rodosek, and H. Reiser, “Advancing the Security of LDACS,” *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 5237–5251, Dec. 2022, doi: 10.1109/TNSM.2022.3189736.

[71] T. Ewert, N. Maurer, and T. Graupl, “Group Key Distribution Procedures for the L-Band Digital Aeronautical Communications System (LDACS),” *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, vol. 2021-October, 2021, doi: 10.1109/DASC52595.2021.9594319.

[72] N. Hegde and S. S. Manvi, “Secure Group Key Management Scheme For Dynamic Vehicular Cloud Computing,” *International Journal of Advanced Networking and Applications*, vol. 13, no. 01, pp. 4821–4826, 2021, doi: 10.35444/IJANA.2021.13103.

[73] J. I. Escribano Pablos and M. I. González Vasco, “Secure post-quantum group key exchange: Implementing a solution based on Kyber,” *IET Communications*, vol. 17, no. 6, pp. 758–773, Apr. 2023, doi: 10.1049/CMU2.12561.

[74] M. Burmester and Y. G. Desmedt, “Efficient and secure conference-key distribution,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 1189, pp. 119–129, 1997, doi: 10.1007/3-540-62494-5_12/COVER.

[75] M. Burmester and Y. Desmedt, “A secure and scalable Group Key Exchange system,” *Inf Process Lett*, vol. 94, no. 3, pp. 137–143, May 2005, doi: 10.1016/J.IPL.2005.01.003.

[76] Y. Desmedt, T. Lange, and M. Burmester, “Scalable authenticated tree based group key exchange for ad-hoc groups,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4886 LNCS, pp. 104–118, 2007, doi: 10.1007/978-3-540-77366-5_12/COVER.

[77] M. Just and S. Vaudenay, “Authenticated multi-party key agreement,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 1163, pp. 36–49, 1996, doi: 10.1007/BFB0034833/COVER.

[78] M. Abdalla, J. M. Bohli, M. I. G. Vasco, and R. Steinwandt, "(Password) authenticated key establishment: From 2-party to group," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4392 LNCS, pp. 499–514, 2007, doi: 10.1007/978-3-540-70936-7_27/COVER.

[79] H. B. Hougaard and A. Miyaji, "Authenticated logarithmic-order supersingular isogeny group key exchange," *Int J Inf Secur*, vol. 21, no. 2, pp. 207–221, Apr. 2022, doi: 10.1007/S10207-021-00549-4/METRICS.

[80] A. Fujioka, K. Takashima, and K. Yoneyama, "One-Round Authenticated Group Key Exchange from Isogenies," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11821 LNCS, pp. 330–338, 2019, doi: 10.1007/978-3-030-31919-9_20/COVER.

[81] S. Furukawa, N. Kunihiro, and K. Takashima, "Multi-party Key Exchange Protocols from Supersingular Isogenies," *Proceedings of 2018 International Symposium on Information Theory and Its Applications, ISITA 2018*, pp. 208–212, Jul. 2019, doi: 10.23919/ISITA.2018.8664316.

[82] K. Takashima, "Post-Quantum Constant-Round Group Key Exchange from Static Assumptions," pp. 251–272, 2021, doi: 10.1007/978-981-15-5191-8_18.

[83] R. Choi, D. Hong, S. Han, S. Baek, W. Kang, and K. Kim, "Design and Implementation of Constant-Round Dynamic Group Key Exchange from RLWE," *IEEE Access*, vol. 8, pp. 94610–94630, 2020, doi: 10.1109/ACCESS.2020.2993296.

[84] D. Apon, D. Dachman-Soled, H. Gong, and J. Katz, "Constant-round group key exchange from the ring-LWE assumption," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11505 LNCS, pp. 189–205, 2019, doi: 10.1007/978-3-030-25510-7_11/COVER.

[85] G. T. Davies, H. Galteland, K. Gjøsteen, and Y. Jiang, "Cloud-Assisted Asynchronous Key Transport with Post-Quantum Security," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12248 LNCS, pp. 82–101, 2020, doi: 10.1007/978-3-030-55304-3_5/COVER.

[86] J. I. E. Pablos, M. I. G. Vasco, M. E. Marriaga, and Á. L. P. Del Pozo, "Compiled Constructions towards Post-Quantum Group Key Exchange: A Design from Kyber," *Mathematics 2020, Vol. 8, Page 1853*, vol. 8, no. 10, p. 1853, Oct. 2020, doi: 10.3390/MATH8101853.

[87] J. I. E. Pablos, M. I. G. Vasco, M. E. Marriaga, and Á. L. P. Del Pozo, "Compiled Constructions towards Post-Quantum Group Key Exchange: A Design from Kyber," *Mathematics 2020, Vol. 8, Page 1853*, vol. 8, no. 10, p. 1853, Oct. 2020, doi: 10.3390/MATH8101853.

[88] M. I. González Vasco, Á. L. Pérez Del Pozo, and R. Steinwandt, "Group Key Establishment in a Quantum-Future Scenario," *Informatica*, vol. 31, no. 4, pp. 751–768, Sep. 2020, doi: 10.15388/20-INFOR427.

[89] J. I. Escribano Pablos and M. I. González Vasco, "Secure post-quantum group key exchange: Implementing a solution based on Kyber," *IET Communications*, vol. 17, no. 6, pp. 758–773, Apr. 2023, doi: 10.1049/CMU2.12561.

[90] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *Symposium on the Theory of Computing*, pp. 84–93, 2005, doi: 10.1145/1060590.1060603.

[91] A. Langlois and D. Stehle, "Worst-Case to Average-Case Reductions for Module Lattices," *Cryptology ePrint Archive*, 2012.

[92] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6110 LNCS, pp. 1–23, 2010, doi: 10.1007/978-3-642-13190-5_1/COVER.

[93] A. Banerjee, C. Peikert, and A. Rosen, "Pseudorandom functions and lattices," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7237 LNCS, pp. 719–737, 2012, doi: 10.1007/978-3-642-29011-4_42/COVER.

[94] J. P. D'Anvers, A. Karmakar, S. Sinha Roy, and F. Vercauteren, "Saber: Module-LWR based key exchange, CPA-secure encryption and CCA-secure KEM," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10831 LNCS, pp. 282–305, 2018, doi: 10.1007/978-3-319-89339-6_16/COVER.

[95] K. Hövelmanns, E. Kiltz, S. Schäge, and D. Unruh, "Generic Authenticated Key Exchange in the Quantum Random Oracle Model," *Cryptology ePrint Archive*, 2018.

[96] K. Hövelmanns, E. Kiltz, S. Schäge, and D. Unruh, "Generic Authenticated Key Exchange in the Quantum Random Oracle Model," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12111 LNCS, pp. 389–422, 2020, doi: 10.1007/978-3-030-45388-6_14/FIGURES/15.

[97] T. Saito, K. Xagawa, and T. Yamakawa, "Tightly-secure key-encapsulation mechanism in the quantum random oracle model," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10822 LNCS, pp. 520–551, 2018, doi: 10.1007/978-3-319-78372-7_17/FIGURES/10.

[98] M. Bellare, A. Boldyreva, and A. O'Neill, "Deterministic and Efficiently Searchable Encryption," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4622 LNCS, pp. 535–552, 2007, doi: 10.1007/978-3-540-74143-5_30.

[99] R. Cramer and V. Shoup, “Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack,” *Cryptology ePrint Archive*, 2001.

[100] J. Nam, J. Paik, and D. Won, “A security weakness in Abdalla et al.’s generic construction of a group key exchange protocol,” *Inf Sci (N Y)*, vol. 181, no. 1, pp. 234–238, Jan. 2011, doi: 10.1016/J.INS.2010.09.011.

[101] “SABER: Mod-LWR based KEM (Round 3 Submission)”, Accessed: Apr. 30, 2024. [Online]. Available: https://www.esat.kuleuven.be/cosic/pqcrypto/saber/

[102] A. Karmakar, J. M. B. Mera, S. S. Roy, and I. Verbauwhede, “Saber on ARM: CCA-secure module lattice-based key encapsulation on ARM,” *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2018, no. 3, pp. 243–266, Aug. 2018, doi: 10.13154/TCHES.V2018.I3.243-266.

[103] M. K. Hasan, Z. Weichen, N. Safie, F. R. A. Ahmed, and T. M. Ghazal, “A Survey on Key Agreement and Authentication Protocol for Internet of Things Application,” *IEEE Access*, vol. 12, pp. 61642–61666, 2024, doi: 10.1109/ACCESS.2024.3393567.

[104] A. Samiullah, F. Akeylek, S. L. Gan, and M. L. Aun, “Group key Management in Resource Constraint Environment: Applications and Use Cases,” *International Journal of Advanced Natural Sciences and Engineering Researches*, vol. 7, no. 3, pp. 269–278, Apr. 2023, doi: 10.59287/IJANSER.419.

[105] R. Cramer and V. Shoup, “Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack,” *IACR Cryptol. ePrint Arch.*, vol. 33, no. 1, pp. 167–226, Nov. 2003, doi: 10.1137/S0097539702403773.